

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO DE FIN DE GRADO

TUTOR DE PROGRAMACIÓN EN FLUTTER

Sergio Rebollo Hernández
Tutor: Alejandro Sierra Urrecho

DICIEMBRE 2020

TUTOR DE PROGRAMACIÓN EN FLUTTER

Autor: Sergio Rebollo Hernández
Tutor: Alejandro Sierra Urrecho

Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
DICIEMBRE 2020

Agradecimientos

En primer lugar, a usted, por interesarse por este trabajo.

A todos los profesores que me han dado clase a lo largo de estos años gracias a los cuales me he formado. En especial a mi tutor Alejandro, por darme la posibilidad de realizar este trabajo, dedicándome su conocimiento, tiempo y ánimo.

Finalmente, a mi familia y amigos, por su apoyo incondicional y por mostrarme el camino correcto.

Abstract

Abstract — Nowadays technology plays an important role in all areas of society. Almost services are offered to customers digitally. The reason of this situation is due to two technological events of the end of the 20th century that have conditioned our lifestyle: computers were distributed for personal and private use; and mobile phones have evolved into what is known today as Smartphones.

Until approximately a decade ago, web services have led the way technologically. Then, Smartphones appeared: devices with a wireless Internet access that help people to fulfill their needs. After that moment, Smartphones have gained much importance versus computers, which makes them a candidate for being the favorite device to perform a wide range of different tasks. This is not surprising because they keep us connected, they provide convenient access to services and they save us time and effort, specific functionalities that we demand today.

In many cases, the services have evolved from web services to mobile applications. In some cases, the two services are kept at same time, for example, WhatsApp, one of the most popular messaging applications, provides users with both a mobile application and a web service. This situation poses a problem for the programmers: they need to adapt the service for each platform in its native language and it may take too much time to develop.

To solve this problem, developers can use Flutter, an open-source SDK library, with which they can develop a 100 % native application for Android, iOS and Web from the same source code.

In this context, the objective of this Degree's Project consists in investigating the viability of developing a project with Flutter, by creating an application from scratch for the education field.

As a result, we will know about the advantages and disadvantages of this SDK library and we will also get a new version of the current Encode's tool: Share&Learn, an application created from scratch and deployed in multi-platform environment.

Key words — Smartphone, Web, Flutter, Android, iOS, Multi-platform, Encode, 100 % native application

Resumen

Resumen — Hoy en día, la tecnología juega un papel fundamental en todos los ámbitos de la sociedad. Prácticamente todos los servicios se ofrecen vía digital. Esto se debe a que, a finales del siglo XX, ocurrieron dos acontecimientos a nivel tecnológico que han marcado nuestro estilo de vida: se comercializaron las computadoras para uso personal y privado; y los teléfonos móviles han evolucionado hasta lo que hoy conocemos como Smartphone.

Hasta hace una década dar un servicio Web ya era ir a la vanguardia en términos tecnológicos. Sin embargo, aparecieron los Smartphones, móviles con un rápido acceso a internet que, para la mayoría de la población, cubría sus necesidades. A partir de este momento, los teléfonos inteligentes han ganado terreno al ordenador, siendo el dispositivo favorito para realizar la mayoría de tareas. No es de extrañar, ya que nos mantienen conectados, facilitan el acceso a los servicios y nos ahorran tiempo y esfuerzo, que es lo que un usuario demanda hoy en día.

En muchos casos, el servicio que se ofrece a la población ha evolucionado de la página Web a la aplicación móvil. Incluso en algunos casos, se mantienen ambos servicios, por ejemplo, WhatsApp, la aplicación de comunicación, que además de una aplicación para móvil cuenta con servicio Web. Esto plantea un problema al desarrollador: crear un servicio para cada plataforma en su lenguaje nativo puede llevar demasiado tiempo de desarrollo.

Para resolver este problema el desarrollador puede hacer uso de Flutter, una librería de código abierto que permite crear una aplicación 100 % nativa para Android, iOS y la Web partiendo de una única base de código.

En este contexto, este Trabajo de Fin de Grado tiene como objetivo investigar la viabilidad de llevar a cabo un proyecto con Flutter, desarrollando una aplicación multiplataforma desde cero para el ámbito de la educación haciendo uso de esta librería.

Como resultado se conocerán los pros y contras de esta librería y además se obtendrá una nueva versión de la herramienta actual Encode, llamada Share&Learn creada desde cero y desplegada en un entorno multiplataforma.

Palabras clave — Smartphone, Web, Flutter, Android, iOS, Multiplataforma, Encode, Aplicación 100 % nativa

Acrónimos

EPS Escuela Politécnica Superior. 2, 3, 6

HTML HyperText Markup Language. 38

IOS iPhone Operating System. 2

JSON JavaScript Object Notation. 26

UAM Universidad Autónoma de Madrid. 1, 12

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Alcance	3
1.4. Metodología	4
1.5. Estructura del documento	4
2. Estado del Arte	5
2.1. Aplicaciones actuales para el aprendizaje	5
2.1.1. Google Codelabs	5
2.1.2. Code School	5
2.1.3. Encode	6
2.1.4. Codecademy	6
2.1.5. Conclusiones	6
2.2. Tecnologías Web, Android e iOS	7
2.2.1. Flutter	8
2.2.2. Xamarin	8
2.2.3. React Native	8
2.2.4. Conclusiones	9
2.3. Algoritmo de repaso espaciado SM2	9
3. Análisis de requisitos	11
3.1. Requisitos funcionales	12
3.2. Requisitos no funcionales	15
4. Diseño	17
4.1. Arquitectura	17
4.1.1. Docente	17
4.1.2. Estudiante	20
4.2. Diseño de la interacción	22
4.2.1. Gestor de un curso	22
4.2.2. Estudiante de un curso	22
4.3. Mapa de navegación de la aplicación	22
4.4. Diseño de la interfaz	23
4.4.1. Wireframes	23
4.5. Diseño de la base de datos	24
5. Desarrollo	27

5.1. Tecnologías y herramientas utilizadas	27
5.2. Implementación de los subsistemas y de la Base de Datos	28
5.2.1. Gestión de base de datos	28
5.2.2. Gestión de usuarios	29
5.2.3. Seguimiento	30
5.2.4. Navegación de contenido	31
5.2.5. Gestión de contenido	31
5.2.6. Banco de preguntas	33
6. Pruebas y resultados	35
6.1. Pruebas de funcionamiento de la aplicación	35
6.1.1. Pruebas sobre la lógica	35
7. Conclusiones y líneas futuras	37
Bibliografía	39
Apéndices	41
A. Estructura del proyecto	43
B. Diagramas de casos de uso	45
B.1. Diagrama de caso de uso de gestión de contenido	45
B.2. Diagrama de caso de uso del banco de preguntas	46
B.3. Diagrama de caso de uso de la navegación de contenido	47
B.4. Diagrama de caso de uso del seguimiento	47
C. Diagramas de flujo	49
C.1. Gestor de un curso	49
C.2. Estudiante de un curso	50
D. Mapa de navegación	51
E. Maquetas	53
E.1. Maquetas Web	53
E.2. Maquetas móvil	61
F. Producto final	77
F.1. Vista Web	77
F.2. Vista móvil	87

Índice de figuras

4.1. Diagrama Entidad Relación de la base de datos local.	25
4.2. Diagrama base de datos NoSQL.	26
B.1. Diagrama de casos de uso de gestión de contenido	45
B.2. Diagrama de casos de uso del banco de preguntas	46
B.3. Diagrama de casos de uso de la navegación de contenido	47
B.4. Diagrama de casos de uso del seguimiento	47
C.1. Diagrama de flujo para el profesor.	49
C.2. Diagrama de flujo para el estudiante.	50
D.1. Mapa de navegación de la herramienta Share&Learn.	51
E.1. Maqueta de vista de inicio de sesión Web.	53
E.2. Maqueta de vista de registro de usuario Web.	54
E.3. Maqueta de vista de la pantalla principal Web.	54
E.4. Maqueta de vista del seguimiento de usuario Web.	55
E.5. Maqueta de vista de pregunta abierta Web.	55
E.6. Maqueta de vista de pregunta tipo test Web.	56
E.7. Maqueta de vista de lista de unidades Web.	56
E.8. Maqueta de vista de objetivos Web.	57
E.9. Maqueta de vista de lista de objetivos Web.	57
E.10. Maqueta de vista de ajuste de cursos Web.	58
E.11. Maqueta de vista de ajuste de unidades Web.	58
E.12. Maqueta de vista de ajuste de objetivos Web.	59
E.13. Maqueta de vista de ajuste de contenido Web.	59
E.14. Maqueta de vista de ajuste de etiquetas Web.	60
E.15. Maqueta de vista de ajuste de preguntas Web.	60
E.16. Maqueta de vista de inicio de sesión móvil.	61
E.17. Maqueta de vista de registro de usuario móvil.	62
E.18. Maqueta de vista de la pantalla principal móvil.	63
E.19. Maqueta de vista del seguimiento de usuario móvil.	64
E.20. Maqueta de vista de pregunta abierta móvil.	65
E.21. Maqueta de vista de pregunta tipo test móvil.	66
E.22. Maqueta de vista de lista de unidades móvil.	67
E.23. Maqueta de vista de objetivos móvil.	68
E.24. Maqueta de vista de lista de objetivos móvil.	69
E.25. Maqueta de vista de ajuste de cursos móvil.	70
E.26. Maqueta de vista de ajuste de unidades móvil.	71

E.27. Maqueta de vista de ajuste de objetivos móvil.	72
E.28. Maqueta de vista de ajuste de contenido móvil.	73
E.29. Maqueta de vista de ajuste de etiquetas móvil.	74
E.30. Maqueta de vista de ajuste de preguntas móvil.	75
F.1. Vista de inicio de sesión Web.	77
F.2. Vista de registro de usuario Web.	78
F.3. Vista de pantalla principal Web.	78
F.4. Vista de seguimiento Web.	79
F.5. Vista de pregunta abierta Web.	79
F.6. Vista de pregunta tipo test Web.	80
F.7. Vista de lista de unidades Web.	80
F.8. Vista de objetivo 1 Web.	81
F.9. Vista de objetivo 2 Web.	81
F.10. Vista de objetivo 3 Web.	82
F.11. Vista de objetivo 4 Web.	82
F.12. Vista de lista de objetivos Web.	83
F.13. Vista de ajuste de curso Web.	83
F.14. Vista de ajuste de unidades Web.	84
F.15. Vista de ajuste de objetivos Web.	84
F.16. Vista de ajuste de contenido: Información y código Web.	85
F.17. Vista de ajuste de contenido: Ejercicio Web.	85
F.18. Vista de ajuste de etiquetas Web.	86
F.19. Vista de ajuste de preguntas Web.	86
F.20. Vista de inicio de sesión móvil.	87
F.21. Vista de registro de usuario móvil.	88
F.22. Vista de pantalla principal móvil.	89
F.23. Vista de seguimiento de usuario móvil.	90
F.24. Vista de pregunta abierta móvil.	91
F.25. Vista de pregunta tipo test móvil.	92
F.26. Vista de lista de unidades móvil.	93
F.27. Vista de ejemplo de objetivo 1 móvil.	94
F.28. Vista de ejemplo de objetivo 2 móvil.	95
F.29. Vista de ejemplo de objetivo 3 móvil.	96
F.30. Vista de ejemplo de objetivo 4 móvil.	97
F.31. Vista de lista de objetivos móvil.	98
F.32. Vista de ajustes de curso móvil.	99
F.33. Vista de ajustes de unidades móvil.	100
F.34. Vista de ajustes de objetivos móvil.	101
F.35. Vista de ajustes de contenido: Información móvil.	102
F.36. Vista de ajustes de contenido: Código móvil.	103
F.37. Vista de ajustes de contenido: Ejercicio móvil.	104
F.38. Vista de ajustes de etiquetas móvil.	105

F.39. Vista de ajustes de pregunta: Abierta móvil.	106
F.40. Vista de ajustes de pregunta: Test móvil.	107

1 | Introducción

El propósito de este Trabajo de Fin de Grado consiste en realizar una investigación de la librería Flutter [1] de Google mediante el desarrollo de una aplicación de apoyo al aprendizaje de programación para la Escuela Politécnica Superior, de la Universidad Autónoma de Madrid (UAM). Este capítulo recoge una descripción de los motivos que han llevado a la realización de este proyecto, sus objetivos principales y una presentación de la estructura de este documento.

1.1. Motivación

Desde las últimas décadas del siglo XX, en el punto de vista tecnológico, han ocurrido dos acontecimientos importantes: primero, la comercialización de las computadoras para uso personal y privado; segundo y más tarde, la evolución de los dispositivos móviles hasta lo que hoy conocemos como Smartphones [2].

Esta tecnología se ha incorporado a nuestras vidas jugando un papel fundamental. Nos permiten un acceso rápido a Internet, nos mantienen “conectados” con el resto de personas, nos facilitan el acceso a los servicios y nos ahorran tiempo y esfuerzo en nuestra vida cotidiana. Debido a estas características, su uso se puede extender al de la docencia, facilitando la adquisición de conocimientos a través de nuevos modelos de aprendizaje.

Algunas ventajas del uso de esta tecnología como modelo de aprendizaje son:

- **Motivación del estudiante.** El uso de un ordenador o un dispositivo móvil en un proceso de aprendizaje frente a una clase convencional puede suscitar mayor interés del estudiante en la materia.
- **No presencial.** El acceso al contenido no dependerá de la Universidad, por lo que el estudiante podrá aprender sin necesidad de estar en el aula. Es un apartado a tener en cuenta de acuerdo con la situación actual provocada por el Covid-19.
- **Adaptación al horario del estudiante.** Muchos estudiantes compatibilizan los estudios con el trabajo u otras actividades, por lo que sus horarios pueden no ser compatibles con los ofrecidos por la universidad. Con el acceso constante a la información el alumno podrá estudiar los contenidos designados por el profesor en el momento que vea adecuado.
- **Adaptación al ritmo del estudiante.** No todos los estudiantes presentan la misma facilidad en el proceso de aprendizaje o tienen las mismas necesidades. A través de estas herramientas el estudiante puede llevar su propio ritmo en la adquisición del conocimiento sin depender del resto de compañeros.

- **Acceso constante al contenido.** La disposición del contenido en la Web o una aplicación puede favorecer el repaso del estudiante siendo esta información completa y correcta.
- **Aprender de manera práctica.** A través de ejercicios, los estudiantes pueden aprender el contenido de manera práctica. Además, las nuevas tecnologías nos permiten desarrollar algoritmos de aprendizaje que en un aula convencional resultaría difícil.
- **Dar soporte multiplataforma.** Permitir al estudiante mantenerse conectado al contenido en cualquier lugar y momento, ya sea a través de un ordenador vía Web o un dispositivo móvil con sistema operativo Android o iPhone Operating System (IOS).

Actualmente, la docencia se lleva a cabo principalmente a través del modelo tradicional, es decir, utilizando libros de texto o planes a través de diapositivas y comprobando los conocimientos del estudiante a través de exámenes. Sin embargo, poco a poco se puede ver como hay una gran oferta de cursos en Internet a través de herramientas online.

Con este Trabajo de Fin de Grado se busca introducir en las aulas un nuevo método de aprendizaje a través de las nuevas tecnologías aprovechando todas sus ventajas. La aplicación Share&Learn tiene las siguientes características:

- **Los estudiantes siempre estarán conectados.** Tendrán un acceso al contenido a través de sus ordenadores vía Web o a través de sus dispositivos móviles con sistema operativo Android o iOS.
- Los estudiantes podrán disponer del contenido de una manera estructurada y sencilla, dividiendo un curso en unidades y cada unidad en objetivos, lo que permitirá al estudiante **aprender de manera incremental**.
- Los estudiantes podrán **asimilar y memorizar el contenido** gracias a ejercicios prácticos. Experimentarán un nuevo método de aprendizaje gracias al algoritmo de repaso espaciado SM2 [3].
- Tanto estudiantes como docentes podrán **crear y gestionar** cursos y ejercicios.

1.2. Objetivos

El objetivo principal de este proyecto consiste en realizar una investigación sobre la posibilidad de extender Encode, aplicación en uso por parte de la Escuela Politécnica Superior (EPS), a un entorno multiplataforma.

Se partirá desde cero en el diseño y desarrollo de una nueva versión de Encode con soporte Web, iOS y Android, gracias a la librería Flutter de Google con lenguaje de

programación Dart, investigando los pros y contras de esta tecnología con vistas a una posible futura puesta en marcha. El contenido de los cursos se obtiene mediante un módulo que carga la información de la aplicación actual.

Además, Share&Learn cubrirá las siguientes funcionalidades:

1. Facilitar a todos los usuarios la creación de cursos de un modo estructurado a través de unidades y objetivos, además de su modificación y eliminación.
2. Facilitar a todos los usuarios la creación, modificación y eliminación de ejercicios para cada unidad de un curso.
3. Almacenar un registro para cada usuario con los cursos de los cuales es propietario, los que está estudiando, las preguntas a resolver, etc.
4. Facilitar a todos los usuarios el acceso al contenido para su estudio, así como la realización de los ejercicios.
5. Permitir a todos los usuarios realizar un seguimiento de su progreso en los ejercicios.
6. Permitir a todos los usuarios un acceso rápido y cómodo desde cualquier dispositivo (Web, Android e iOS).

1.3. Alcance

El público al que se destina este Trabajo de Fin de Grado son los docentes y estudiantes de la EPS. Sin embargo, el desarrollo de la aplicación se ha hecho genérica, permitiendo su uso en otros niveles del sistema educativo.

Share&Learn es una aplicación multiplataforma para Web, Android e IOS que busca ayudar a los estudiantes en el proceso de aprendizaje de programación, así como a los docentes en la creación y gestión de los cursos. De esta manera gracias a la aplicación los estudiantes podrán avanzar en los contenidos y servirá de refuerzo junto con las clases convencionales en aula.

El uso de la librería de código abierto Flutter puede incentivar a los docentes a tomar la decisión de desarrollar una nueva versión de Encode. No solo por dar soporte multiplataforma, sino por beneficiarse de la facilidad de esta tecnología, así como las funcionalidades que ofrece, creando una nueva versión que mejore la usabilidad, interfaz y portabilidad; así como incluir nuevos requisitos.

1.4. Metodología

Tras analizar el proyecto a desarrollar, se ha determinado que la metodología ágil más conveniente para Share&Learn sigue un ciclo de vida incremental e iterativo.

Para el desarrollo del proyecto se han establecido cinco etapas:

1. Análisis.
2. Diseño.
3. Codificación.
4. Pruebas.
5. Instauración y aprobación del sistema.

Los incrementos se realizarán en el siguiente orden:

1. **Iteración 1:** Subsistema de gestión de usuarios.
2. **Iteración 2:** Subsistema de gestión de contenido.
3. **Iteración 3:** Subsistema banco de preguntas.
4. **Iteración 4:** Subsistema de navegación de contenido.
5. **Iteración 5:** Subsistema de seguimiento.

1.5. Estructura del documento

- En el **capítulo 2** se describe el estado del arte introduciendo al lector en la situación actual de aplicaciones existentes y tecnologías de desarrollo multiplataforma.
- En el **capítulo 3** se especifica el alcance de la aplicación y la metodología aplicada en el desarrollo de este proyecto.
- En el **capítulo 4** se detalla el catálogo de requisitos funcionales y no funcionales.
- En el **capítulo 5** se describe el diseño de la aplicación.
- En el **capítulo 6** se describe el proceso de desarrollo de la aplicación.
- En el **capítulo 7** se detallan las pruebas realizadas durante y tras el desarrollo de la aplicación.
- En el **capítulo 8** se presentan las conclusiones obtenidas tras del desarrollo de la aplicación en lo referente a la librería Flutter de Google.

2 | Estado del Arte

Este capítulo expone un análisis de algunas herramientas actuales similares a la de este Trabajo de Fin de Grado. Dicho análisis incluye la descripción de sus ventajas e inconvenientes, enmarcando este Trabajo de Fin de Grado en su contexto.

2.1. Aplicaciones actuales para el aprendizaje

Hasta principios de la segunda década de este siglo, el modelo establecido de aprendizaje comprendía la presencialidad del estudiante con un plan de enseñanza general.

Sin embargo, en los últimos años, la facilidad de alcance de la tecnología al igual que el acceso a Internet, junto con la iniciativa de muchos usuarios a emprender el aprendizaje de nuevos conocimientos han desembocado en la proliferación de herramientas que permiten a cada usuario aprender por su cuenta. De este modo, no es necesaria la presencialidad del estudiante en un curso, además de que puede adaptar su tiempo aprendiendo a su propio ritmo.

A continuación, se muestran algunas de las herramientas actuales.

2.1.1. Google Codelabs

Google Codelabs [4] es una herramienta web desarrollada por Google que sirve para crear y publicar cursos paso a paso. Los usuarios pueden aprender con estos tutoriales, de manera guiada y práctica; ya que cada curso se compone de una serie de pasos que el usuario debe completar. El método de aprendizaje se basa en crear pequeñas aplicaciones, o añadir funcionalidad a otras ya existentes, de modo que sirva de base al usuario para el tema que está estudiando. Todos los pasos cuentan con solución, de manera que el usuario puede comprobar o corregir su desarrollo. Estos cursos están abiertos a todo el público, sin necesidad de registro.

2.1.2. Code School

Code School [5] es una plataforma web que ofrece cursos en formato de vídeo con ejercicios basados en la codificación. Además, con el fin de ayudar a los usuarios en el proceso de aprendizaje, da soporte para asistencia en vivo a través de videollamada,

mensajería o uso compartido de pantalla. El acceso al contenido de esta plataforma requiere de registro y pago.

2.1.3. Encode

Encode [6] es una aplicación web utilizada por la EPS. A través de ella imparten tres cursos, completando las clases convencionales en aula, por cursos online paso a paso. Cada curso se divide en una serie de unidades que a su vez se divide en objetivos. De esta manera, el estudiante aprende de manera guiada e incremental a través de los objetivos y unidades. Cada objetivo, cuenta con tres apartados: información, código y ejercicio. Con ello, se consigue que el estudiante centre su atención en el objetivo, aprendiendo por pasos y afianzando el contenido gracias al código de muestra y los ejercicios a resolver. A fin de que el estudiante memorice los contenidos a largo plazo, la aplicación utiliza el algoritmo de repaso espaciado SM2 para presentar preguntas al estudiante en relación a las unidades que previamente ha estudiado.

2.1.4. Codecademy

Codecademy [7] es una plataforma web que ofrece cursos online. Cada curso se divide en una serie de pasos que el usuario debe realizar para completarlo. En cada paso, se exponen unos contenidos, y para comprobar que el usuario los comprende, se le pide que realice un ejercicio de codificación. Si el ejercicio ha sido resuelto correctamente, el usuario podrá continuar con el siguiente paso, en caso contrario, recibe información acerca del error que ha cometido. Para motivar a los usuarios, esta plataforma ofrece foros de discusión y puntúa al usuario en función de los ejercicios realizados, generando una clasificación con el resto. En esta plataforma, el acceso al contenido requiere de registro y pago.

2.1.5. Conclusiones

Las cuatro herramientas anteriores se basan en el mismo principio a la hora de presentar el contenido al usuario: trata de dividir un curso en pasos u objetivos, de modo que el usuario aprende de manera incremental. Además, como método de aprendizaje, por cada paso, la aplicación pide al usuario la realización de un pequeño ejercicio, lo que le sirve como método de auto-evaluación. Las diferencias son las siguientes:

- **Google Codelabs** busca que el usuario aprenda de manera práctica realizando una pequeña aplicación o modificando una ya existente. En cada paso, pedirá una de estas modificaciones, hasta que al final del curso el usuario consigue una aplicación funcional.
- El método de **Codecademy** o **Code School** es distinto al de Google Codelabs. Esta herramienta pide al usuario realizar pequeños ejercicios de codificación que

tienen que ver con el paso en que se encuentra. De este modo, cuando el usuario termine el curso, no tendrá una aplicación como en Google Codelabs, pero si un conocimiento práctico de todos los pasos del curso.

- **Encode** utiliza un método que permite al estudiante memorizar a largo plazo el contenido del curso. La evaluación del usuario sobre el contenido engloba el método de Google Codelabs, y Codecademy y Code School. Es decir, además de realizar poco a poco una aplicación que finalmente será funcional, evalúa con pequeñas preguntas de cada paso al usuario. Además, para que el usuario memorice estos contenidos a largo plazo utiliza el algoritmo de repaso espaciado SM2 para presentar preguntas al estudiante en relación con los pasos u objetivos que previamente ha estudiado.

Otro apartado a destacar es la sofisticación de las herramientas Codecademy o Code school, que ofrecen más funcionalidades como por ejemplo asistencia particular en el caso de Code School o foros de discusión en Codecademy. Esto se debe a que son herramientas que requieren un pago mensual o anual.

La herramienta que se desarrolla en este proyecto reúne las principales características positivas que se han identificado en las herramientas presentadas, facilitando tanto la gestión como la presentación de cursos online. Las principales características son:

- **Aprendizaje incremental.** Divide una tarea en subtarear. Cada curso, se dividirá en unidades, que a su vez se dividirán en objetivos.
- **Plataforma colaborativa.** Cada usuario podrá crear su propio curso, gestionando tanto su contenido como sus ejercicios.
- **Aprendizaje basado en algoritmo de repaso espaciado SM2.** El usuario memorizará el contenido rápida y fácilmente.
- **Multiplataforma.** Acceso vía Web y aplicaciones para Android e iOS.

2.2. Tecnologías Web, Android e iOS

Hace ya una década que se comenzó a fabricar y extender los primeros dispositivos móviles inteligentes [8], lo que supuso pasar de un teléfono cuya única finalidad era realizar llamadas a “conectar” al usuario con el resto de personas al igual que facilitar el acceso a los servicios. La mejora de estos dispositivos ha causado que en algunos casos se pongan al nivel de un ordenador en términos de prestaciones. Son muchas las personas que han sustituido el uso del ordenador por el de un dispositivo móvil en su vida cotidiana.

Por estas razones, cuando hoy día se planifica un proyecto, es importante pensar en su despliegue en la Web, pero también en crear aplicaciones para Android e iOS, que son los principales sistemas operativos de los Smartphone [9]. De esta manera, facilitamos el acceso al servicio desde casi cualquier plataforma.

Realizar un desarrollo nativo para cada una de estas plataformas resulta costoso, ya que supone desarrollar la misma aplicación tres veces. La búsqueda de un método que facilite el desarrollo para estas tres plataformas ha causado la aparición de librerías como Flutter, que ha sido la utilizada para llevar a cabo este proyecto. Además, podemos encontrar otros ejemplos como Xamarin o React Native.

2.2.1. Flutter

Flutter [10] es una librería de código abierto de desarrollo multiplataforma orientado al rendimiento, interfaz y experiencia de usuario. Fue desarrollada por Google con una primera versión estable a finales de 2018. El lenguaje de programación utilizado es Dart, desarrollado también por Google, y que sirve tanto para la interfaz de usuario como la lógica de la aplicación. Permite desarrollar aplicaciones para Android, iOS y la Web a partir de una misma base de código. La principal ventaja es que genera código 100% nativo para cada plataforma y así obtiene un mayor rendimiento.

2.2.2. Xamarin

Xamarin [11] es una librería de código abierto presentada en 2011 y adquirida por Microsoft en 2016. Permite desarrollar aplicaciones para Android e IOS con C Sharp como lenguaje de programación. Como característica específica destaca que permite realizar llamadas a código desarrollado para una plataforma (Java en Android o Swift en IOS). De esta manera, se puede reutilizar código específico de una plataforma o desarrollar funciones comunes a ambas. En el desarrollo de la interfaz, Xamarin permite desarrollar componentes que se mostrarán igual en ambas plataformas o componentes nativos específicos.

2.2.3. React Native

React Native [12] es otra de las librerías de código abierto utilizadas en la actualidad. Desarrollada por Facebook, permite crear aplicaciones para Android e iOS. Para el desarrollo de la lógica de la aplicación se utiliza JavaScript como lenguaje de programación, y para la interfaz React, una librería desarrollada por Facebook para la creación de interfaz de usuario. React se encarga de traducir al lenguaje nativo de la plataforma en que se ejecute la aplicación. En React Native encontramos un componente llamado Bridge que se encarga de la comunicación entre la parte de la lógica y la de la interfaz.

2.2.4. Conclusiones

Flutter ofrece varias mejoras y ventajas en comparación con Xamarin y React Native, lo que ha llevado a la decisión de utilizar esta librería. Dichas mejoras y ventajas son las siguientes:

- Aumenta la productividad del desarrollador gracias a la opción Hot Reload, que le permite visualizar los cambios de interfaz al momento sin necesidad de reiniciar la aplicación.
- Cuenta con una librería de interfaz más completa y actual gracias a Material [13] y Cupertino [14].
- Permite el desarrollo de aplicaciones Web.
- Fue lanzada en 2018, fecha reciente, por lo que se encuentra actualmente en pleno crecimiento y con perspectiva de futuro de crear aplicaciones para Windows, macOS, Linux o sistemas embebidos (Smartwatch) [15].
- Genera código 100 % nativo para cada plataforma en que se despliega.

2.3. Algoritmo de repaso espaciado SM2

Como se ha detallado en apartados anteriores, la aplicación que se desarrolla en este proyecto utiliza el algoritmo de repaso espaciado SM2 para presentar preguntas al estudiante en relación con los objetivos que previamente ha estudiado, de modo que le resulte más sencillo memorizar los contenidos a largo plazo. A este tipo de aplicaciones se las denomina "SRS" (Spaced Repetition Software) [16] ya que usan el algoritmo de repaso espaciado para que el usuario aprenda rápida y fácilmente un determinado contenido.

Esta técnica se basa en el efecto espaciado, que es un fenómeno psicológico por el cual un individuo puede memorizar información fácilmente a base de repetir información aprendida en varias ocasiones y a lo largo de un período de tiempo. Por tanto, se trata de una técnica simple que consiste en repasar información en repetidas ocasiones pero en intervalos de tiempo cada vez más largos a medida que dicha información se ha memorizado con éxito.

Las aplicaciones que incorporan la técnica SM2 presentan una pregunta al usuario y éste debe recordar su respuesta. Una vez que el usuario ha respondido, el programa le pide que evalúe cuánto le costó recordar la respuesta, y así calcular el tiempo en volver a mostrar dicha pregunta. Además, este algoritmo utiliza retroalimentación, es decir, realiza el cálculo del tiempo en función de la evaluación actual y de anteriores para una misma pregunta. Por ejemplo, para la aplicación Share&Learn el usuario evaluará cuanto le costó recordar la respuesta con tres opciones: fácil (no le costó), dudo (le costó, pero lo recordó)

y difícil (no supo la respuesta). En caso de ser una pregunta tipo test, el proceso será automático: si acierta la respuesta se establecerá como fácil y si falla como difícil.

A pesar de que la técnica resulta sencilla, eso no quiere decir que no sea efectiva, todo lo contrario. El cerebro humano se deshace rápidamente de la información que no le resulta útil. Hay estudios que certifican que olvidamos el 75 % del material que aprendemos tras un periodo de 48 horas [17]. La solución a esta situación es simple: repasar.

3 | Análisis de requisitos

Este capítulo expone el análisis de requisitos funcionales y no funcionales del proyecto. El conjunto de estos requisitos definen las características finales de la aplicación Share&Learn.

El catálogo de requisitos se encuentra dividido en los siguientes subsistemas:

- **Gestión de usuarios:** Subsistema encargado de dar soporte al registro e inicio de sesión de los usuarios.
- **Gestión de contenido:** Subsistema encargado de dar soporte a la creación, eliminación y modificación de los cursos.
- **Banco de preguntas:** Subsistema encargado de dar soporte a la creación, eliminación y modificación de preguntas.
- **Navegación de contenido:** Subsistema encargado de dar soporte a la visualización de los cursos y resolución de sus preguntas.
- **Seguimiento:** Subsistema encargado de dar soporte al algoritmo de repaso espaciado SM2 para las preguntas de los usuarios.

3.1. Requisitos funcionales

Gestión de usuarios

- RF 1.** Crear cuentas para gestionar usuarios.
- RF 2.** Distinguir entre tres tipos de usuarios: no matriculados en ningún curso, matriculados en un curso y propietarios de un curso.
 - RF 2.1.** Para los tres tipos de usuario, ofrecer la posibilidad de crear un nuevo curso; convirtiéndolo en su propietario.
 - RF 2.2.** Para los tres tipos de usuario, ofrecer la posibilidad de matricularse en un curso, siempre que no sean propietarios.
 - RF 2.3.** Si el usuario está matriculado en un curso, ofrecer la posibilidad de acceso a la información de las distintas unidades que contiene, así como la resolución de sus ejercicios.
 - RF 2.4.** Si el usuario es propietario de un curso, ofrecer la posibilidad de modificar o eliminar dicho curso. Además, se le ofrecerá acceso al contenido de sus unidades, pero sin posibilidad de realizar sus ejercicios.
- RF 3.** Requerir el nombre, apellidos, cuenta de correo institucional de la UAM y contraseña del usuario en el registro de usuarios.
- RF 4.** Permitir la identificación del usuario mediante un formulario que solicita el correo y la contraseña con la que previamente se ha registrado.

Gestión de contenido

- RF 5.** La estructura de contenidos establece una jerarquía: cursos -> unidades -> objetivos.
- RF 6.** Permitir la creación o eliminación de cursos.
- RF 7.** El proceso de creación de un curso establece un nombre y descripción por defecto. Permitir modificar dichos campos.
- RF 8.** Permitir la creación o eliminación de múltiples unidades para un curso.
- RF 9.** El proceso de creación de una unidad establece un nombre por defecto. Permitir modificar dicho campo.
- RF 10.** Permitir modificar el orden de las unidades de un curso.
- RF 11.** Permitir la creación o eliminación de múltiples objetivos para una unidad.
- RF 12.** El proceso de creación de un objetivo establece un nombre por defecto. Permitir modificar dicho campo.

- RF 13.** Permitir modificar el orden de los objetivos de una unidad.
- RF 14.** Además del nombre, un objetivo puede estar formado por: una o ninguna pestaña de información, múltiples o ninguna pestañas de código y múltiples o ninguna pestañas de ejercicio.
- RF 14.1.** Permitir la creación o eliminación de la pestaña de información, así como la modificación de su contenido.
- RF 14.2.** Permitir la creación o eliminación de múltiples pestañas de código, así como la modificación de sus contenidos.
- RF 14.3.** Permitir la creación o eliminación de múltiples pestañas de ejercicios, así como la modificación de sus contenidos.
- RF 14.3.1.** Distinguir entre dos tipos de ejercicios: enunciado o pregunta tipo test.
- RF 14.3.1.1.** Para las preguntas tipo test, permitir la creación o eliminación de múltiples opciones, así como la selección de la opción correcta.

Banco de preguntas

- RF 15.** Permitir la creación o eliminación de múltiples etiquetas para una unidad.
- RF 16.** El proceso de creación de una etiqueta establece un nombre por defecto. Permitir modificar dicho campo.
- RF 17.** Permitir la creación o eliminación de múltiples preguntas para una etiqueta.
- RF 18.** Distinguir entre dos tipos de preguntas: tipo abierta y tipo test.
- RF 18.1.** La creación de una pregunta tipo abierta establece un enunciado y respuesta por defecto, permitir modificar dichos campos.
- RF 18.2.** La creación de una pregunta tipo test establece un enunciado y una opción por defecto, permitir modificar dichos campos. Permitir crear o eliminar múltiples opciones, así como la selección de la opción correcta.
- RF 19.** Permitir establecer el número máximo de preguntas a realizar para una determinada unidad, de acuerdo con las preguntas de las etiquetas que contiene.

Estudio de cursos

- RF 20.** Mostrar al usuario en la pantalla principal un listado con los cursos matriculados.
- RF 21.** Mostrar las unidades del curso seleccionado previamente por el usuario.
- RF 22.** Mostrar el objetivo de la unidad seleccionada previamente por el usuario.
- RF 23.** El acceso a unidades y objetivos es secuencial.

- RF 23.1.** Una unidad u objetivo se desbloquea con la finalización del anterior, salvo la primera unidad y primer objetivo que están desbloqueados automáticamente tras la matriculación en el curso.
- RF 24.** Mostrar para su realización las preguntas de las etiquetas de una unidad al finalizar esta por primera vez.
 - RF 24.1.** Aplicar el algoritmo de repaso espaciado SM2 a la pregunta tras la respuesta del usuario a ésta.
 - RF 24.2.** Guardar las preguntas en el repositorio de preguntas del usuario.
- RF 25.** Mantener un repositorio por cada usuario con las preguntas de las unidades que desbloquea.

Seguimiento

- RF 26.** Mostrar al usuario las preguntas programadas de acuerdo con el algoritmo de repaso espaciado SM2.
 - RF 26.1.** Aplicar el algoritmo de repaso espaciado SM2 a la pregunta tras la respuesta del usuario a ésta.
 - RF 26.2.** Guardar las preguntas en el repositorio del usuario.
- RF 27.** Mostrar al usuario información de interés relativa al algoritmo de repaso espaciado SM2.
 - RF 27.1.** Mostrar el número de preguntas pendientes por responder.
 - RF 27.2.** Mostrar el intervalo medio de las preguntas respondidas hasta el momento.
 - RF 27.3.** Mostrar la facilidad media de las preguntas respondidas hasta el momento.

3.2. Requisitos no funcionales

Usabilidad

- RNF 1.** El aprendizaje del uso de la aplicación no debe requerir un gran esfuerzo.
- RNF 2.** El sistema debe ser efectivo, de modo que los usuarios logren sus objetivos (aprendizaje de los cursos).
- RNF 3.** El sistema debe ser consistente, de modo que tareas semejantes se realicen de forma parecida.
- RNF 4.** Una tarea estará dividida en subtareas de modo que se facilite al usuario llevar a cabo acciones en la aplicación.
- RNF 5.** Las distintas pantallas contarán con mecanismos de vuelta a la pantalla anterior.

Interfaz

- RNF 6.** El texto de la aplicación se adaptará al lenguaje del navegador o del teléfono, exceptuando el contenido de los cursos que dependerá del escogido por el propietario.
- RNF 7.** Las distintas pantallas contarán con un diseño amigable y adaptativo de acuerdo con la acción que el usuario este realizando.
- RNF 8.** A fin de garantizar la consistencia entre las distintas pantallas se utilizarán guías de estilo.
- RNF 9.** El texto será corto, claro y con estilos similares.

Rendimiento

- RNF 10.** El tiempo de respuesta de la aplicación no debe exceder los 5 segundos, de modo que se garantice rapidez en la interacción usuario-aplicación.

Portabilidad

- RNF 11.** La información quedará almacenada en una base de datos Cloud Firestore.

Seguridad lógica y de datos

- RNF 12.** El acceso a la aplicación requerirá la identificación del usuario mediante correo y contraseña. Solo podrán acceder a ella introduciendo los datos correctamente.
- RNF 13.** A fin de proteger la información de la base de datos, las contraseñas estarán cifradas.

Requisitos técnicos

RNF 14. La aplicación se podrá ejecutar en navegadores como Chrome, o en teléfonos con sistema operativo Android e iOS.

4 | Diseño

En este capítulo se detalla el diseño de la aplicación, basado en el análisis realizado en el capítulo anterior.

4.1. Arquitectura

La arquitectura de la aplicación se divide en dos partes: docentes y estudiantes. La parte de los docentes comprende la gestión de los cursos: su creación, eliminación o modificación de contenido (unidades, objetivos y ejercicios o preguntas). En cuanto a la parte de los estudiantes, comprende la navegación por los cursos para llevar a cabo su estudio y el seguimiento de su progreso, que conlleva responder a sus preguntas pendientes.

La aplicación Share&Learn permite a un docente navegar por cursos llevando a cabo las tareas de un estudiante, y al igual ocurre con los estudiantes, que pueden crear sus propios cursos y gestionarlos. Esto quiere decir que un mismo usuario puede tener dos roles en la aplicación según las acciones que lleve a cabo. Sin embargo, a fin de simplificar el análisis que se expone a continuación, se tendrá en cuenta que el docente siempre actuará como docente y el estudiante como estudiante.

A continuación, se describe cada subsistema organizando la información según el rol que tenga el usuario en la aplicación.

4.1.1. Docente

Las tareas que un docente lleva a cabo están relacionadas con la gestión de los cursos y preguntas, que corresponden con los subsistemas Gestión de contenido y Banco de preguntas.

Gestión de contenido

Este subsistema tiene que ver con la gestión de la información de la aplicación, es decir, de los cursos que ofrece. Engloba las siguientes opciones:

Creación/Eliminación/Modificación de cursos

La creación de un curso es una operación muy sencilla. Una vez el docente ha iniciado sesión tiene la opción de crear un curso pulsando un botón en la pantalla principal. El curso creado tiene un nombre y descripción por defecto y no cuenta con ninguna unidad.

Con el curso ya creado el docente tiene tres opciones: modificar el nombre y descripción, crear unidades para dicho curso y eliminarlo.

Creación/Eliminación/Modificación de unidades

Para llevar a cabo estas acciones, el docente debe navegar hasta los ajustes de unidades del curso. La operación de creación de una unidad es igual a la de un curso, pulsando un botón se crea una unidad con un nombre por defecto y con ningún objetivo.

Con la unidad ya creada el docente tiene 3 opciones: modificar el nombre, crear objetivos para dicha unidad y modificar el orden de las unidades.

El proceso de reordenación de unidades resulta sencillo e intuitivo. En los ajustes de unidades, éstas se muestran en una lista reordenable, es decir, los elementos de dicha lista pueden reordenarse arrastrándolos a la posición deseada. Las unidades se incorporan al final de la lista a medida que se crean.

Creación/Eliminación/Modificación de objetivos

Para llevar a cabo estas acciones, el docente debe navegar hasta los ajustes de objetivos de una unidad. Ocurre lo mismo que con la creación de un curso o unidad, pulsando un botón se creará un objetivo con un nombre por defecto, y en este caso, con las pestañas de información, código y ejercicio desactivadas. Desactivadas quiere decir que cuando el usuario visualice el objetivo encontrará una pantalla en blanco, ya que primero hay que activar las pestañas e introducir la información deseada. Esta operación de activación y desactivación permite que un objetivo tenga las pestañas que el gestor desee, por ejemplo, un objetivo con pestañas de información y código, tendrán dichas pestañas activadas y la de ejercicio desactivada. Equivale a hacer visible u ocultar las pestañas.

Con el objetivo creado el docente puede: modificar el nombre y la información de las pestañas (inicialmente vacías), activarlas o desactivarlas y modificar el orden de los objetivos.

Modificar la información de las pestañas es sencillo, el usuario la introduce en un campo de texto. Para activar o desactivar una pestaña el docente pone a “ON” u “OFF” un “switch” respectivamente para cada una. Cabe destacar que la pestaña de ejercicio permite crear una pregunta (u ejercicio) solo con enunciado, o una pregunta tipo test con las opciones que se deseen.

En cuanto a la reordenación de objetivos, es exactamente igual que en el caso de las unidades, se trata de una lista reordenable.

En la Figura B.1 se muestra el diagrama de casos de uso para la gestión de contenido para los docentes.

Banco de preguntas

Este subsistema tiene que ver con la creación de preguntas en relación con las unidades de un curso. Engloba las siguientes opciones:

Creación/Eliminación/Modificación de etiquetas

Una etiqueta es una agrupación de preguntas bajo un nombre (etiqueta) en relación con un tema. Por ejemplo, la etiqueta “kotlin_if” es una agrupación de preguntas en relación a la instrucción “if” del lenguaje de programación Kotlin. Esta jerarquía se lleva a cabo para que la gestión de las preguntas sea más sencilla y ordenada, de este modo un mismo tema puede tener varias etiquetas y cada etiqueta un conjunto de preguntas.

Por tanto, para poder crear una etiqueta, debe haber un curso con unidades ya creadas. Con este requisito, el docente ya puede crear las etiquetas que desee navegando hasta la pantalla de gestión de etiquetas (la etiqueta tendrá un nombre por defecto).

Posteriormente, el usuario podrá modificar el nombre de la etiqueta, eliminarla o crear sus preguntas. Como opción adicional el docente puede seleccionar el número de preguntas a realizar para el usuario que estudie la unidad correspondiente. Por ejemplo, el docente selecciona 30 preguntas, entonces la aplicación mostrará las 30 primeras preguntas de la unidad, teniendo en cuenta todas las etiquetas y sus preguntas correspondientes.

Creación/Eliminación/Modificación de preguntas

Para llevar a cabo estas acciones, el usuario debe haber creado una etiqueta previamente. Existen dos tipos de preguntas: tipo abierta con enunciado y solución por defecto, y tipo test con enunciado y una opción por defecto.

En cuanto a la modificación, para una pregunta tipo abierta, el usuario puede modificar el campo del enunciado o de la solución; y en las tipo test, puede modificar el enunciado, crear o eliminar opciones y modificar sus campos, además de marcar la opción correcta.

Por último, el docente puede eliminar las preguntas.

En la figura B.2 se muestra el diagrama de casos de uso para el banco de preguntas, que representa los pasos que deben llevarse a cabo para las distintas acciones.

4.1.2. Estudiante

Navegación de contenido

Este subsistema permite a los estudiantes la visualización de los cursos, así como de sus unidades y objetivos. Engloba las siguientes operaciones:

Visualización de cursos

La pantalla principal de la aplicación Share&Learn está dividida en tres partes: cursos en los que se ha matriculado el usuario, cursos de los cuáles el usuario es el gestor y cursos disponibles en los que no está matriculado ni es propietario.

De este modo, el usuario tiene una visión global del estado de todos los cursos en la aplicación, y además, puede: matricularse en cursos, acceder a los ajustes de uno de sus cursos o acceder a los temas de los cursos en los que está matriculado.

Visualización de unidades

Una vez el estudiante selecciona uno de los cursos en los que está matriculado, la aplicación muestra el listado de unidades de dicho curso. De este modo, el usuario podrá acceder a los objetivos de una unidad seleccionándola en la lista, siempre y cuando haya cursado la unidad anterior (salvo la primera unidad que siempre es accesible).

Visualización de objetivos

Para visualizar un objetivo, el usuario debe seleccionar la unidad a la que pertenece. En este caso, la aplicación muestra las pestañas de dicho objetivo (aquellas que el propietario haya activado). Como una unidad puede contener más de un objetivo, una vez se selecciona la unidad, la aplicación muestra el último objetivo al que el usuario accedió.

Para pasar de un objetivo a otro, se pincha sobre un botón “Siguiente” en la última pestaña del objetivo. Si es el último objetivo de la unidad, la aplicación muestra la pantalla de visualización de unidades, y es el usuario quien debe seleccionar la siguiente.

En la figura B.3 se muestra el diagrama de casos de uso para la visualización de cursos por parte del estudiante.

Seguimiento

Este subsistema da soporte al algoritmo SM2. Permite guardar las preguntas que el usuario ha ido respondiendo y presentárselas en la fecha adecuada.

Por cada unidad que el estudiante finaliza, la aplicación muestra una serie de preguntas, las cuáles se guardan para llevar a cabo el algoritmo de repaso espaciado.

El subsistema de seguimiento le permite al usuario realizar las siguientes acciones:

Visualizar su valoración

El usuario puede ver su valoración en relación con las preguntas que responde: número de preguntas pendientes por responder, intervalo medio (representa el número de días medio en realizar una misma pregunta) y la facilidad media (representa la dificultad que tiene el usuario a responder las preguntas que se le presentan).

Responder las preguntas pendientes

El estudiante podrá responder a las preguntas que tiene pendientes.

En la figura B.4 se muestra el diagrama de casos de uso para el seguimiento.

4.2. Diseño de la interacción

En este apartado se presentan los diagramas de flujo con el fin de mostrar la interacción de los usuarios con la aplicación. Para ello, se distinguen dos tipos de usuarios: el gestor de un curso (por ejemplo, un docente) y el que estudia un curso (por ejemplo, un estudiante).

4.2.1. Gestor de un curso

El gestor de un curso es aquel que lo ha creado y ha administrado su contenido (unidades, objetivos, ejercicios y preguntas). En la figura C.1 se puede ver en detalle las opciones que ofrece la aplicación para el gestor y su funcionalidad correspondiente.

4.2.2. Estudiante de un curso

El estudiante de un curso es aquel que navega por su contenido con la finalidad de aprenderlo sin opción de modificarlo ni gestionarlo. En la figura C.2 se puede ver en detalle las opciones que ofrece la aplicación para el estudiante y su funcionalidad correspondiente.

4.3. Mapa de navegación de la aplicación

Con el fin de cumplir con la funcionalidad anteriormente descrita, así como su interacción, se desarrollará una aplicación que contará con varias pantallas.

La figura D.1 muestra el mapa de navegación de la aplicación Share&Learn, ofreciendo una visión global de las distintas pantallas, así como sus conexiones.

Como se puede observar, se distinguen tres colores: el azul, representa que dichas pantallas son accesibles a todos los usuarios, el rojo, representa las pantallas que son solo accesibles para el propietario del curso y el morado, representa las pantallas que son solo accesibles para aquellos usuarios que no son propietarios del curso.

4.4. Diseño de la interfaz

Con el fin de cumplir los requisitos de interfaz (sección 3.2) y para que la interacción usuario-aplicación sea satisfactoria, la interfaz ha sido diseñada de acuerdo con las pautas de la Guía práctica para el diseño de la interacción con el usuario [18].

Share&Learn tiene como objetivo desplegarse en un entorno Web, Android e iOS. Por esta razón, el diseño de la interfaz debe ser apto para dichas plataformas sin verse comprometido por el entorno en que se despliega, cumpliendo los requisitos técnicos de la sección 3.2.

Share&Learn busca ayudar a estudiantes en el proceso de aprendizaje de programación, así como a los docentes en la creación y gestión de los cursos. Por esta razón, el diseño se centra en el usuario, buscando ser amigable, sencillo y cuyo comportamiento sea intuitivo.

4.4.1. Wireframes

De acuerdo con las especificaciones del apartado anterior, teniendo en cuenta los requisitos de interfaz, los requisitos técnicos y la Guía práctica para el diseño de la interacción con el usuario [18], se han diseñado las maquetas de la aplicación, tanto para el entorno Web como para el entorno móvil, ya sea Android o iOS.

Las maquetas representan una referencia visual del tipo de elemento que aparece en cada página, así como su distribución y la organización de la información.

En el Apéndice E se muestran las maquetas que se han diseñado para entorno Web y móvil.

4.5. Diseño de la base de datos

Para la persistencia de información se han diseñado dos bases de datos:

1. Base de datos SQLite [19]. Se trata de una base de datos relacional que permite al usuario guardar la información de forma local, es decir, en la plataforma a través de la cuál accede a la aplicación Share&Learn. Para ello se ha utilizado el paquete Moor de Flutter [20].
2. Base de datos NoSQL [21]. Se trata de una base de datos no relacional que permite al usuario guardar la información en la nube desde la plataforma a través de la cuál accede a la aplicación Share&Learn. Para ello se ha utilizado el servicio Cloud Firestore de Google [22].

Estas dos bases de datos trabajan sincronizadas para garantizar los servicios de Share&Learn al usuario. Los motivos que llevan a requerir dos bases de datos son los siguientes:

- **Reducir el tráfico de datos.** Inicialmente todos los datos se encuentran en la base de datos NoSQL. Cuando un usuario inicia sesión en una plataforma descarga el contenido de la base de datos NoSQL, guardando la información en la base de datos SQLite. De este modo, las consultas del usuario serán a través de la base de datos local. Una vez que el usuario cierre sesión, la operación se realiza en modo inverso, es decir, la información de la base de datos SQLite se actualiza en la base de datos de la nube.

Con el proceso anterior se consigue un doble efecto, reducir el tráfico de datos a través de Internet que en muchos casos se asocia a costes monetarios con el proveedor de la base de datos en la nube, y aumentar la velocidad de respuesta de la aplicación al depender en mayor medida de la base de datos local.

- **Portabilidad.** Acceso desde cualquier plataforma al realizar una descarga inicial de la información desde la base de datos NoSQL. De este modo, el usuario puede acceder a su perfil vía Web, Android e iOS.

La base de datos SQLite se ha diseñado de acuerdo con el siguiente modelo relacional:

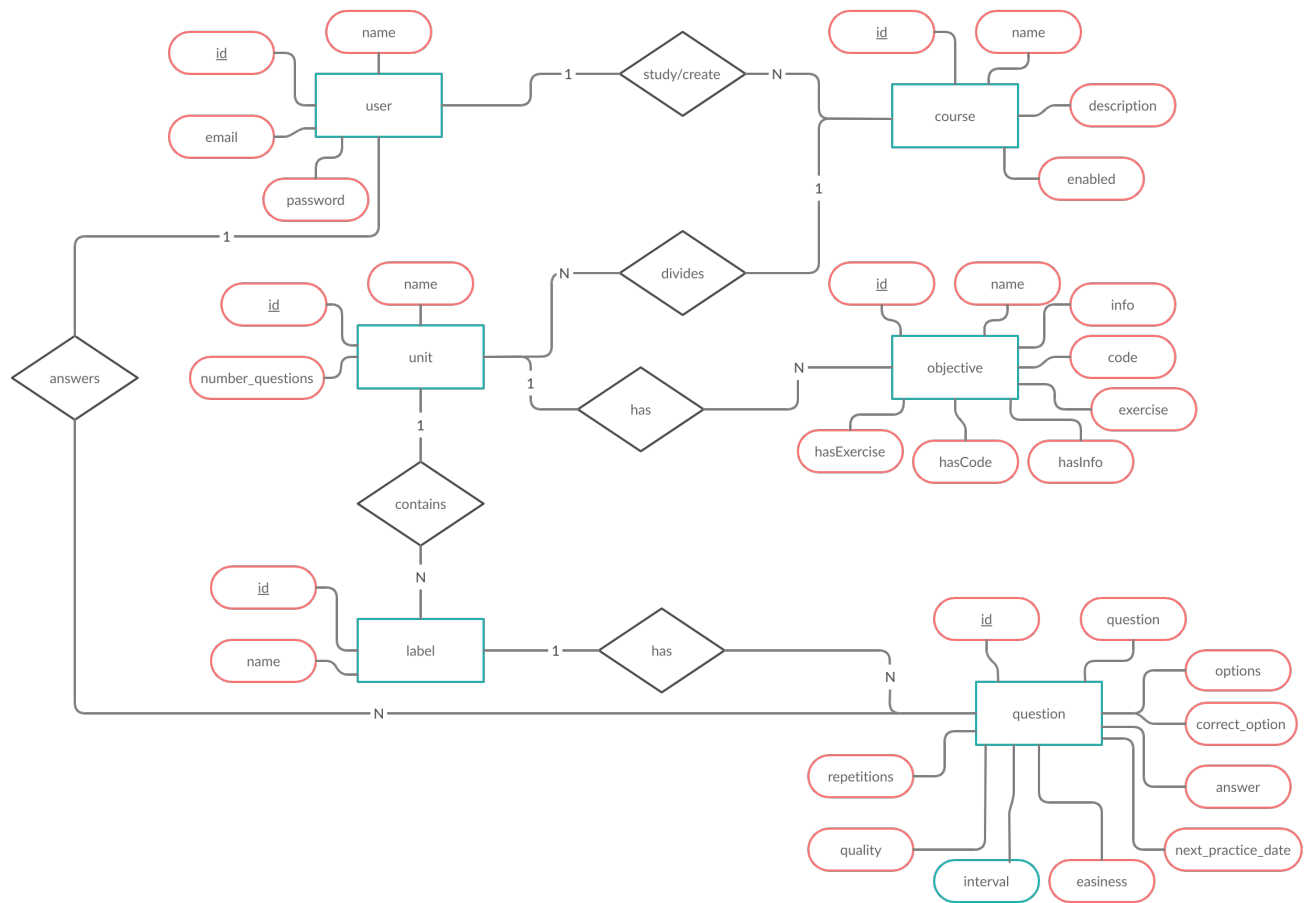


Figura 4.1: Diagrama Entidad Relación de la base de datos local.

En la figura 4.1 se muestra el diagrama Entidad-Relación compuesto por 6 entidades: User, Course, Unit, Objective, Label, Question.

- La entidad **User** representa a los usuarios. Dicha tabla almacena la información referente a los usuarios registrados: nombre de usuario, correo electrónico y contraseña encriptada.
- La entidad **Course** gestiona los datos relacionados con los cursos: nombre, descripción y un atributo de campo booleano (`enabled`) que indica si el curso está abierto al resto de usuarios.
- La entidad **Unit** almacena los datos referentes a las unidades: nombre y número máximo de preguntas a realizar al usuario en dicha unidad.
- La entidad **Objective** gestiona los datos relacionados con los objetivos: nombre, información, código y ejercicio (3 pestañas que forman un objetivo) y tres campos booleanos que nos indican la presencia de las pestañas.

- La entidad **Label** almacena los datos referentes a las etiquetas de preguntas: nombre de la etiqueta.
- La entidad **Question** gestiona los datos relacionados con las preguntas: enunciados de la pregunta, opciones (si es tipo test), opción correcta (si es tipo test), respuesta y los atributos para el algoritmo de aprendizaje espaciado SM2 (intervalo, repeticiones, facilidad, calificación y fecha de próxima presentación).

Para la relación entre entidades se utilizan identificadores (id).

Como se ha especificado anteriormente, para la base de datos en la nube se ha utilizado el servicio Cloud Firestore de Google. Cloud Firestore es una base de datos NoSQL que almacena los datos en documentos cuyo contenido tiene formato JavaScript Object Notation (JSON). Estos documentos a su vez se almacenan en colecciones. Este modelo permite una gran flexibilidad al poder almacenar objetos anidados complejos siempre de manera ordenada y jerárquica. [21]

Un ejemplo de dicho modelo es el siguiente:

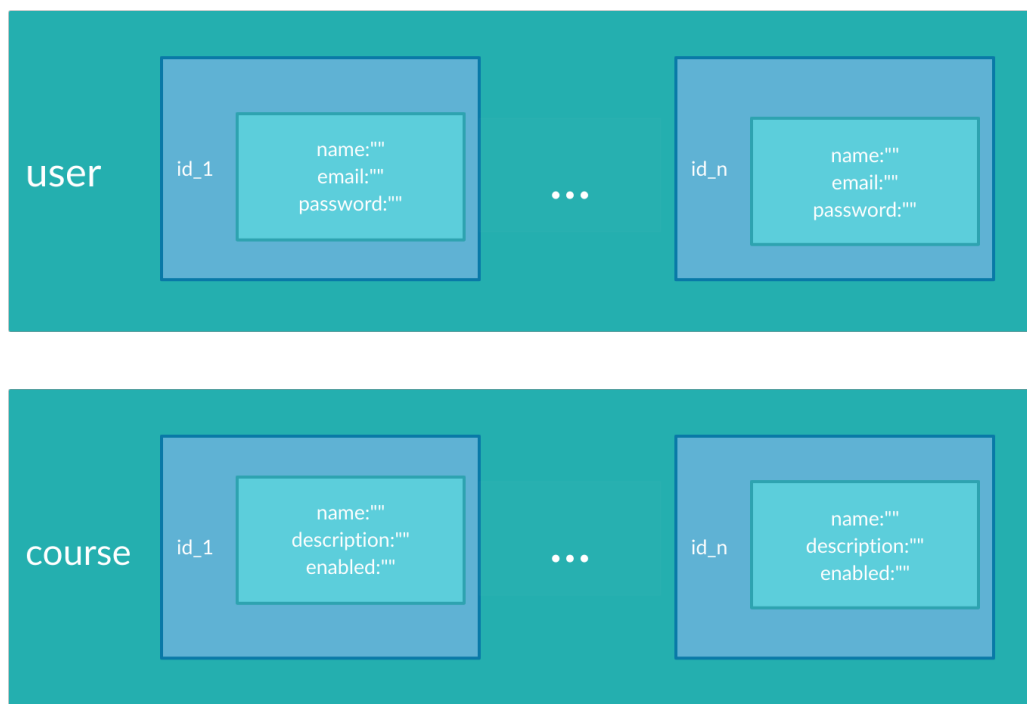


Figura 4.2: Diagrama base de datos NoSQL.

En la figura 4.2 se muestran un ejemplo de las colecciones User y Course, que almacenan la información de los usuarios y de los cursos respectivamente.

Por ejemplo, en el caso de la colección User, se puede ver como los documentos incluidos en esta colección se identifican mediante un atributo del usuario que es un identificador único. Para cada documento, los atributos de la clase User se guardan en formato JSON.

5 | Desarrollo

La aplicación desarrollada se ha basado en el diseño expuesto en el capítulo anterior, mejorando los distintos prototipos hasta llegar al producto final. En concreto, el primer prototipo se basó en la creación de maquetas con el fin de identificar los requisitos. Este primer prototipo se desarrolló y modificó hasta llegar a la versión actual de la aplicación.

La estructura del proyecto Share&Learn está descrita en el Apéndice A.

5.1. Tecnologías y herramientas utilizadas

Para el desarrollo de la aplicación se ha hecho uso de la siguiente tecnología:

- **Flutter** como librería principal para el desarrollo de la aplicación en entornos Android, iOS y web. Desarrollado por Google, tiene como principal ventaja que genera código 100 % nativo para cada plataforma, con lo que el rendimiento es totalmente idéntico al de aplicaciones nativas tradicionales.
- **Dart** [23] como lenguaje principal para la codificación tanto de interfaz como para el tratamiento de datos. Esto se debe a que el desarrollo de una aplicación mediante la librería Flutter se puede realizar íntegramente en Dart.
- **Material Design de Google** [13] como librería para la definición de la interfaz de la aplicación. Fue desarrollada por Google y está incluida en Flutter para el diseño de interfaces. Incluye todo lo referente a iconos, tipografías, botones, cuadros, menús, etc.
- **Moor** para la persistencia de datos local. Permite tanto el almacenamiento como la gestión de datos en Dart, lo que facilita su uso e incorporación en un proyecto Flutter. Se trata de una librería que permite la creación de una base de datos SQL que funciona en entornos Android, iOS y en la Web.
- **Cloud Firestore** para el almacenamiento de los datos en la nube y la sincronización entre distintos dispositivos. Permite crear una base de datos NoSQL y desplegarla en entornos Android, iOS y Web, accediendo a la información desde cualquier dispositivo.

Las herramientas que se han utilizado durante el desarrollo han sido las siguientes:

- **AndroidStudio** [24] como entorno de desarrollo y programación.

- Firebase Console para la gestión de la base de datos NoSQL en la nube.
- Balsamiq [25] para la realización de las maquetas de la aplicación.
- Google Drive [26] para el control de versiones.
- Creately [27] para el diseño de los diagramas de las bases de datos, los diagramas de caso de uso, de flujo y de interacción.
- LateX Project [28] como software para la elaboración del presente documento.

5.2. Implementación de los subsistemas y de la Base de Datos

A continuación, se describe cómo se ha implementado cada uno de los subsistemas. En el Apéndice F se muestran las distintas vistas de la aplicación.

5.2.1. Gestión de base de datos

A pesar de que este apartado no describe un subsistema, es necesario para aclarar cuestiones acerca de las dos bases de datos implementadas y su sincronización.

Como se ha descrito en capítulo 4, para la persistencia de información se ha decidido implementar dos bases de datos: una SQL local para la persistencia y tratamiento de información cuando el usuario se encuentra en la sesión, y otra NoSQL en la nube para permitir el acceso y sincronización de la información desde cualquier dispositivo.

Sea cual sea el entorno por el que se conecta el usuario a la aplicación (Android, iOS o web), se realiza una descarga de la información de la base de datos de la nube y se vuelca sobre la base de datos local. De esta manera, el usuario tiene la información actualizada e interactuará con la base de datos de su dispositivo (local). Durante la sesión todas las operaciones son sobre la base de datos y cuando se realiza un cambio se actualiza además en la base de datos de la nube.

Por tanto, el primer paso en la ejecución de la aplicación es bajar la información de la base de datos de la nube e insertarla en la local.

Otro dato de interés, importante para las consultas de la base de datos, es que todos los “objetos” de la aplicación están identificados mediante un identificador, lo que facilita las operaciones “SELECT”. Por ejemplo, para realizar un “SELECT” de los cursos, se haría una consulta “SELECT id FROM COURSE”, donde id representa el identificador del curso a buscar.

5.2.2. Gestión de usuarios

Cuando un usuario accede por primera vez a la aplicación debe registrarse a través de un formulario. El usuario debe rellenar tres campos solicitados: nombre de usuario, correo electrónico y contraseña. A continuación, pulsará el botón “Crear cuenta” y la aplicación realizará las siguientes operaciones:

1. Comprobar que los tres campos no estén vacíos.
2. Comprobar que el campo de correo electrónico esté introducido correctamente mediante una expresión regular.

Si las operaciones resultan satisfactorias, se cifra la contraseña mediante Hash MD5 [29] y se realiza una consulta a la base de datos local, lo que conlleva las siguientes operaciones:

1. Comprueba que no exista un usuario con el correo electrónico especificado. Si la operación “SELECT COUNT” devuelve 0.
2. De ser satisfactoria la operación anterior, se inserta la información en la base de datos mediante un “INSERT” y se presenta la pantalla principal al usuario.

En el caso de que el usuario ya se haya registrado previamente, deberá autenticarse para acceder a la aplicación. Para ello, el usuario debe rellenar un formulario con campos de correo electrónico y contraseña. La contraseña se cifra mediante Hash MD5 y se realiza una consulta a la base de datos local que conlleva las siguientes operaciones:

1. Se comprueba si existe un usuario con el correo electrónico especificado, y si existe se obtiene. Esta operación es un “SELECT USER”.
2. Si la operación anterior ha sido satisfactoria, se comprueba que la contraseña cifrada obtenida de la base de datos sea la misma que la cifrada que ha introducido el usuario.

Si la segunda operación es satisfactoria se presenta la pantalla principal al usuario.

Cuando un usuario descarga la aplicación en su móvil, puede que no quiera realizar la operación de autenticación cada vez que desee utilizarla, para ello, se han implementado preferencias.

Se trata de un fichero local que guarda información como por ejemplo si el usuario se ha autenticado previamente, en cuyo caso no será necesario el proceso de autenticación. De este modo, el usuario que descargue la aplicación en su móvil solo necesitará el registro inicial.

Esta funcionalidad corresponde con las pantallas de las figuras F.1 y F.2 para Web, y F.20 y F.21 para móvil.

5.2.3. Seguimiento

Este subsistema tiene como fin mostrar al usuario información de interés respecto al algoritmo SM2 de aprendizaje espaciado, mostrar las preguntas pendientes y además permitir responderlas.

Cuando el usuario navega a la pantalla de seguimiento, se realizan las siguientes operaciones de consulta en la base de datos:

1. Se obtiene el usuario de la base de datos mediante su identificador “SELECT id FROM USER”.
2. Una vez obtenida esa información se procede al tratamiento de los datos.

Las preguntas del usuario se guardan en formato JSON, lo que permite guardar el enunciado, la solución o los datos del algoritmo SM2 como una única cadena, y así guardarla en la base de datos. Se convierte de formato JSON a cadena “List<String>” y posteriormente cada elemento de la lista se convierte en un tipo “QUESTION”. A partir de este momento para cada pregunta se realiza lo siguiente:

1. Obtener el valor EASINESS para posteriormente calcular la media de todas las preguntas.
2. Obtener el valor INTERVAL para posteriormente calcular la media de todas las preguntas.
3. Obtener el valor NEXT PRACTICE DATE para saber qué preguntas puede responder el usuario por estar pendientes (fecha de próxima respuesta inferior a la fecha actual).

A partir de esa información: se muestra la facilidad media, el intervalo medio y las preguntas pendientes. Además, se presenta un botón de enlace para responder esas preguntas pendientes.

Cuando se responde a una pregunta se presentan dos escenarios:

1. Se responde por primera vez la pregunta. Para ello, se busca en la cadena JSON de preguntas del usuario si existe alguna con el mismo id. De este modo, se sabe que no existe y es la primera vez, y se añade dicha pregunta a la cadena JSON.
2. Si no es la primera vez, se obtiene la pregunta de la cadena JSON y se actualizan sus datos.

Una vez actualizada la información, es necesario realizar un “UPDATE” en la base de datos para actualizar la lista de preguntas.

Esta funcionalidad corresponde con las pantallas de las figuras F.4 para Web y F.23 para móvil.

5.2.4. Navegación de contenido

Este subsistema permite la visualización de los cursos. Cuando un usuario accede a la pantalla principal de la aplicación se realizan dos operaciones en la base de datos:

1. Se obtienen todos los cursos “SELECT * FROM COURSE”.
2. Se obtiene la información del usuario “SELECT id FROM USER”. De este modo, obtenemos los cursos del usuario, tanto de los que es propietario como en los que está matriculado. (El usuario cuenta con dos campos: cursos de los cuales es propietario y cursos en los que está matriculado, que guardan los identificadores de dichos cursos.)

Aclaración: el campo cursos matriculados es un mapa que contiene como clave el id de un curso y como valores otro mapa, cuya clave es el id de los temas del curso y sus valores los id de sus objetivos. De este modo, se tiene un control de los cursos, unidades y objetivos que el usuario ha desbloqueado.

Con esta información recogida, se crean tres secciones en la pantalla: sección “estudiando” (cursos en los que el usuario está matriculado), sección “cursos disponibles” (aquellos en los que no está matriculado) y “mis cursos” (aquellos que ha creado y gestiona). Los cursos se distribuyen en estas secciones dependiendo de los campos mencionados en la operación 2.

Cuando el usuario accede a un curso matriculado, se muestra la lista de temas disponibles. Para ello se realiza una consulta en la base de datos obteniendo los temas del curso seleccionado. Un tema estará disponible para un usuario si el id de dicho tema se encuentra en el campo cursos matriculados del usuario. (Se busca que uno de los valores para el id del curso sea el id del tema). Esta operación es la misma para mostrar los objetivos, que solo se muestran si se encuentran en el mapa, y a medida que se desbloquean unidades u objetivos se introducen en dicho mapa.

Esta funcionalidad corresponde con las pantallas de las figuras F.7 a la F.11 para Web, y F.26 a la F.30 para móvil.

5.2.5. Gestión de contenido

Este subsistema permite al usuario la gestión de contenido, pudiendo crear cursos, unidades y objetivos. La pantalla principal de la aplicación cuenta con un botón flotante que permite al usuario crear un curso de manera automática. Cuando el usuario presiona dicho botón se realizan las siguientes acciones:

1. Se crea un curso con valores por defecto, por ejemplo, el nombre del curso será “Nombre por defecto” o la lista de unidades de dicho curso estará vacía. Este mismo proceso ocurre siempre que se crea un curso, unidad u objetivo, se establecen sus campos a valores por defecto, que posteriormente pueden ser modificados.

2. Una vez se ha creado el curso, se realiza un “INSERT” en la tabla de cursos de la base de datos.
3. La pantalla se recarga de manera automática y el curso aparecerá en la sección “mis cursos”.

Con el curso ya creado tenemos tres opciones: Pantalla de ajustes de curso.

- **Modificar el nombre y descripción del curso**, así como su visibilidad para el resto de usuarios. Esto conlleva una operación “UPDATE” sobre la base de datos.
- **Eliminar el curso**. Conlleva una operación “DELETE” sobre la base de datos. Esta operación llevaría al usuario de la pantalla de ajustes de curso a la pantalla principal.
- **Crear unidades para dicho curso**. Este proceso conlleva navegar a la pantalla ajustes de unidades.

El proceso de creación de una unidad es el mismo que de un curso, la aplicación crea un objeto “unidad” con sus valores por defecto posteriormente lo inserta en la base de datos mediante un “INSERT”. Para ello el usuario debe navegar a la pantalla de ajustes de unidad, donde se mostrarán las unidades en una “ReorderableList”, que consiste en una lista donde cada elemento es una unidad, representada por su nombre y que se pueden ordenar arrastrando dichos elementos a la posición deseada.

Con la unidad creada tenemos cuatro opciones:

- **Modificar el nombre de la unidad**. Esto conlleva una operación “UPDATE” sobre la base de datos.
- **Eliminar la unidad**. Conlleva una operación “DELETE” sobre la base de datos.
- **Reordenar las unidades**. Para ello arrastraremos los elementos de la lista. Esto conlleva una operación “UPDATE” sobre la base de datos.
- **Crear objetivos para dicho curso**. Este proceso conlleva navegar a la pantalla ajustes de objetivos.

El proceso de creación de objetivos, así como sus posibles operaciones son exactamente igual que el proceso de creación de unidades y sus posibles operaciones, lo que lo hace un proceso más intuitivo de cara al usuario. Sin embargo, entre las operaciones de modificación de objetivos existe la posibilidad de modificar las pestañas. Para ello, el usuario navega hasta la pantalla ajustes de objetivo y tiene las siguientes opciones:

- **Activar/Desactivar las pestañas** mediante un botón “switch”, lo cual implica un “UPDATE” en la base de datos.

- **Modificar el contenido de las pestañas**, que simplemente consiste en rellenar los campos de texto con su correspondiente “UPDATE” en la base de datos.

Esta funcionalidad corresponde con las pantallas de las figuras F.13 a la F.17 para Web, y F.32 a la F.37 para móvil.

5.2.6. Banco de preguntas

Este subsistema permite crear y gestionar las preguntas relacionadas con los temas. El primer paso consiste en crear una etiqueta, que se trata de una agrupación de preguntas bajo un nombre (etiqueta). Para poder crear una etiqueta es necesario haber creado una unidad, ya que las etiquetas se establecen por unidades.

En la pantalla “ajuste de etiquetas” se presenta al usuario una lista donde cada elemento representa una unidad. Cada unidad está compuesta por: el nombre de la unidad y tantos “inputs” como etiquetas tiene dicha unidad, además de un botón para crear una etiqueta. Por tanto, cuando se crea una etiqueta se realiza lo siguiente:

1. Se crea una etiqueta pulsando el botón “+” en la unidad deseada.
2. Se crea un objeto “etiqueta” con los valores por defecto y se realiza un “UPDATE” sobre la tabla “etiquetas” de la base de datos.
3. Se muestra en el elemento de la lista un input con el nombre de la etiqueta.

Introduciendo el nombre deseado en ese input se modificaría el nombre de la etiqueta (“UPDATE”). Para gestionar las preguntas hay que pulsar el icono “ajustes” para navegar a los ajustes de preguntas.

Las posibles operaciones son:

- **Crear una pregunta abierta**, en cuyo caso solo se puede modificar el enunciado y la respuesta.
- **Crear una pregunta de tipo test**, en cuyo caso se puede modificar el enunciado, crear posibles respuestas modificando su contenido y seleccionar la opción correcta.

Esta funcionalidad corresponde con las pantallas de las figuras F.18 a la F.19 para Web, y F.38 a la F.40 para móvil.

6 | Pruebas y resultados

Este capítulo tiene como objetivo describir los procesos de prueba que se han llevado a cabo durante el desarrollo del proyecto, así como los resultados finales obtenidos.

6.1. Pruebas de funcionamiento de la aplicación

A fin de comprobar el correcto funcionamiento de la aplicación se han implantado una serie de pruebas durante su desarrollo.

Se han establecido dos tipos de pruebas, de verificación y de validación, con el fin de desarrollar una aplicación acorde a sus especificaciones y que cumple las necesidades de los usuarios.

6.1.1. Pruebas sobre la lógica

Este conjunto de pruebas busca comprobar que el comportamiento del software es el esperado. Para ello, se han realizado inspecciones de código, pruebas de caja blanca y pruebas de caja negra.

Inspecciones de código

El objetivo de esta fase de pruebas consiste en evaluar el código implementado en busca de posibles defectos. Para llevar a cabo este conjunto de pruebas de una manera ordenada y evitando acumular errores a medida que avanza el desarrollo, se decidió que dicha revisión de código se realizaría al final de cada iteración coincidiendo con el desarrollo de un nuevo subsistema. De esta manera, al comienzo de la siguiente iteración los posibles fallos han sido solventados.

Sin embargo, al añadir nueva funcionalidad se modificaba código existente, por lo que resultó necesario revisar código ya implementado. Por ejemplo, durante la tercera iteración (Subsistema de banco de preguntas) se modificó código de la segunda iteración (Subsistema de gestión de contenido).

Pruebas de caja blanca

El objetivo de esta fase de pruebas consiste en comprobar que el código implementado cumple correctamente su funcionalidad.

Para llevar a cabo esta fase, se diseñó una serie de pruebas unitarias para cada subsistema desarrollado. De nuevo, al igual que con la inspección de código, estas pruebas se han realizado al finalizar el desarrollo del subsistema a fin de no acumular errores y asegurarse de realizar el desarrollo de manera correcta e incremental.

Pruebas de caja negra

El objetivo de esta fase de pruebas consiste en comprobar que la salida del programa a una determinada entrada es correcta, sin importar el procedimiento llevado a cabo.

Para ello, no solo se introducen datos habituales, sino también datos incorrectos para comprobar la salida del programa. De este modo se comprueba que la aplicación hace lo que tiene que hacer y no realiza acciones inesperadas.

Para ello se han llevado a cabo casos con valores límite, pruebas con las bases de datos, pruebas con situaciones de excepción, pruebas de interacción, etc.

7 | Conclusiones y líneas futuras

Para finalizar este trabajo, en este capítulo se exponen las conclusiones obtenidas del desarrollo de la aplicación Share&Learn y algunas ideas para mejorar este proyecto.

Con el desarrollo de la aplicación Share&Learn se ha cumplido con el objetivo principal de este proyecto: realizar una investigación de la librería Flutter de Google mediante el desarrollo de una aplicación de apoyo al aprendizaje de programación para la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

Share&Learn ha cumplido con todos los requisitos impuestos. Facilita al estudiante el proceso de aprendizaje reforzando los conceptos aprendidos en el aula, adecuándose a su tiempo y ritmo, permitiendo que memorice los conceptos de manera incremental y estructurada (gracias a la división de los cursos en unidades y objetivos, y práctica gracias a los ejercicios junto al repaso espaciado). Además, le mantiene “conectado” al contenido, con un acceso rápido y sencillo ya sea por la aplicación para Android o iOS, o vía Web, motivándole y generando mayor interés por el uso de las nuevas tecnologías. Para los docentes, se ha diseñado una interfaz más sencilla y amigable que les ayuda en el proceso de creación y gestión de cursos, así como de ejercicios y preguntas; lo que hace menos tedioso estos procesos.

El desarrollo desde cero de una aplicación mediante la librería Flutter de Google ha permitido conocer las ventajas, así como las limitaciones de esta nueva herramienta para su uso en un proyecto real.

Las ventajas principales son:

- **Desarrollo multiplataforma.** Con el desarrollo de una misma base de código es posible desplegar la aplicación en tres plataformas: Android, iOS y Web. Tiene un doble efecto sobre el desarrollador: reduce el tiempo de desarrollo de la aplicación y además no obliga a conocer el código nativo de estas plataformas, por lo que solo es necesario aprender Dart, que es el lenguaje utilizado por esta librería.
- **Aumenta la productividad.** Flutter cuenta con una opción llamada Hot Reload, por lo que permite actualizar los cambios de interfaz sin necesidad de reiniciar la aplicación. Lo que reduce el tiempo de diseño de las interfaces.
- **Librería de diseño.** Esta librería cuenta con opciones de interfaz amplias y actuales, que en muchos casos están ya predefinidas para el uso del desarrollador, lo que permite crear una interfaz usable y amigable de una manera muy rápida. Resulta muy sencillo diseñar la interfaz de las pantallas obteniendo un resultado muy satisfactorio, que además se adapta correctamente a las distintas plataformas.

- **Facilidad de desarrollo de Backend.** Dart es un lenguaje de programación orientado a objetos que resulta muy parecido a Java. La creación de objetos, así como el acceso a sus atributos en las distintas pantallas resulta sencillo, sin necesidad de aprender el lenguaje en profundidad.
- **Perspectivas de futuro.** Las perspectivas de futuro en cuanto al despliegue de aplicaciones en plataformas como Windows, macOS, Linux o sistemas embebidos puede incentivar el uso de esta librería.

Las limitaciones principales son:

- **Lanzada en 2018.** Esta es la limitación principal, se trata de una librería reciente que se encuentra en crecimiento, lo que hace que no sea tan completa como otras librerías similares.
- **Despliegue Web reciente.** En 2019 se anunció la posibilidad de desplegar las aplicaciones en plataforma Web. Durante el desarrollo del proyecto se encontraron muchas limitaciones en relación con este tema, ya que muchos paquetes estaban creados para despliegue Android e iOS pero no Web, por lo que obligó a buscar paquetes que cumplieran el despliegue en las tres plataformas.
- **Base de datos local.** Google ofrece servicios para bases de datos en la nube que se pueden usar para las tres plataformas, sin embargo, a la hora de buscar un paquete que permitiese crear una base de datos local resultó difícil encontrar uno que se ajustará a la condición de multiplataforma. Existen muchos paquetes de bases de datos locales para Android e iOS pero no incluyen para la Web y la mayoría se estaban actualizando para cumplir con esta última plataforma.

Como conclusión, respondiendo a la pregunta de si es posible usar Flutter en un proyecto real, la respuesta es sí. Para llevar a cabo el desarrollo de un proyecto para Android e iOS, Flutter es una buena opción, sin embargo, de cara a un despliegue Web, todavía es mejor esperar. Por ejemplo, empresas como la española Aures Tic utilizan esta librería para llevar a cabo sus desarrollos.

Se han considerado las siguientes ideas como trabajo futuro:

- Incluir una opción “recuperar contraseña” en el formulario de registro o inicio de sesión.
- Añadir un editor HTML HyperText Markup Language (HTML) que permita personalizar el texto de la información de los objetivos. Por ejemplo, incluir letras en negrita o cursiva, encabezados, distintos tipos de fuente, etc.
- Permitir incluir imágenes en el texto de los objetivos directamente desde el dispositivo que ejecute la aplicación.
- Realizar pruebas con usuarios con vistas a una futura puesta en marcha de la aplicación.

Bibliografía

- [1] *Flutter, información general*. URL: https://flutter.dev/?gclid=CjwKCAiA-f78BRBbEiwATKRRBMEf6VEyrq4flpeFlnRg7bGSKlxyxsXLXGlyQvSv8opa--jYa_hG1RoC8JQQAuD_BwE&gclidsrc=aw.ds.
- [2] *Aparición Smartphone*. URL: <https://www.nextu.com/blog/el-antes-y-despues-del-smartphone/>.
- [3] *Repaso espaciado*. URL: <https://www.genbeta.com/a-fondo/como-el-software-de-repaso-espaciado-puede-ayudarte-a-aprender-idiomasyo-lo-que-te-propongas>.
- [4] *Google Codelabs*. URL: <https://codelabs.developers.google.com/>.
- [5] *Code School*. URL: https://www.pluralsight.com/codeschool?aid=7010a000002BWqGAAW&promo=&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=EMEA_Dynamic&utm_content=&gclid=CjwKCAiA-f78BRBbEiwATKRRBGWCWBSLZqgKRiH82xXtQ7-EkC2LkWz8IOY9KtXZ9FgvmL1BMG2JSBoCyHQQAvD_BwE.
- [6] *Encode*. URL: <https://encode.eps.uam.es/>.
- [7] *Code Academy*. URL: <https://www.codecademy.com/>.
- [8] *Evolución Smartphone*. URL: <https://www.vanessaestorach.com/la-evolucion-de-los-telefonos-moviles/>.
- [9] *Software Smartphone*. URL: <https://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html>.
- [10] *Flutter, características principales*. URL: <https://www.qualitydevs.com/2019/07/05/que-es-flutter/>.
- [11] *Xamarin*. URL: <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>.
- [12] *React Native*. URL: <https://reactnative.dev/>.
- [13] *Material Design*. URL: <https://material.io/develop/flutter>.
- [14] *Cupertino*. URL: <https://flutter.dev/docs/development/ui/widgets/cupertino>.
- [15] *Flutter, perspectiva de futuro*. URL: <https://www.empleosit.com.ar/news/52/%C2%BFES-Flutter-el-framework-del-futuro-.html>.
- [16] *Spaced Repetition Software*. URL: <https://collegeinfo geek.com/spaced-repetition-memory-technique/>.
- [17] *Curva del olvido*. URL: <https://www.eleconomista.es/ecoaula/noticias/9594629/12/18/Olvidamos-el-90-de-lo-que-aprendemos-y-queremos-cambiar-esta-dinamica.html>.

- [18] *Guia práctica para el diseño de la interacción con el usuario*. URL: <http://di002.edv.uniovi.es/~alguero/eaac/Ensenianza/GUIAP.pdf>.
- [19] *Base de datos SQLite*. URL: <https://www.oracle.com/es/database/what-is-a-relational-database/>.
- [20] *Moor*. URL: <https://moor.simonbinder.eu/>.
- [21] *Base de datos NoSQL*. URL: <https://sites.google.com/site/basesdedatosavanzadaute/bases-de-datos-y-bigdata>.
- [22] *Cloud Firestore*. URL: <https://firebase.google.com/docs/firestore?hl=es>.
- [23] *Dart*. URL: <https://dart.dev/>.
- [24] *Android Studio*. URL: <https://developer.android.com/studio?gclid=CjwKCAiA-f78BRBbEiwATKRRBN6Vs0ujHNpSiL47fJm7F7HtILgaZyR99Sq6tRUt8jjmctQbDJ9kAxoCMYMbwE&gclidsrc=aw.ds>.
- [25] *Balsamiq*. URL: <https://balsamiq.cloud/>.
- [26] *Google Drive*. URL: https://www.google.com/intl/es_es/drive/.
- [27] *Creately*. URL: <https://creately.com/>.
- [28] *LateX Project*. URL: <https://www.overleaf.com/>.
- [29] *Hash MD5*. URL: <https://pub.dev/packages/crypto>.

Apéndices

A | Estructura del proyecto

Aquí se presenta el árbol que describe la estructura del proyecto especificando la jerarquía de los directorios desarrollados, excluyendo las bibliotecas externas utilizadas y nombradas anteriormente.

```
/Share&Learn
├── assets
│   ├── json
│   │   ├── android.json
│   │   └── kotlin.json
│   ├── logo
│   │   ├── shareandlearn.png
│   │   └── shareandlearnlogo.png
│   └── strings
│       ├── en.json
│       └── es.json
└── lib
    ├── database
    │   └── databaseHelper.dart
    ├── language
    │   └── appLocalizations.dart
    ├── objects
    │   ├── course.dart
    │   ├── exercise.dart
    │   ├── objective.dart
    │   ├── unit.dart
    │   ├── user.dart
    │   └── sm2
    │       ├── label.dart
    │       └── question.dart
```

```
/
├── lib
│   ├── screens
│   │   ├── listCourses.dart
│   │   ├── listObjectives.dart
│   │   ├── listUnits.dart
│   │   ├── login.dart
│   │   ├── showObjective.dart
│   │   ├── showQuestion.dart
│   │   ├── userInformation.dart
│   │   └── edition
│   │       ├── editCourse.dart
│   │       ├── editLabels.dart
│   │       ├── editListObjectives.dart
│   │       ├── editObjectiveContent.dart
│   │       ├── editQuestions.dart
│   │       └── editUnits.dart
│   └── src
│       ├── main.dart
│       └── arguments.dart
└── pubspec.yaml
```

B | Diagramas de casos de uso

B.1. Diagrama de caso de uso de gestión de contenido

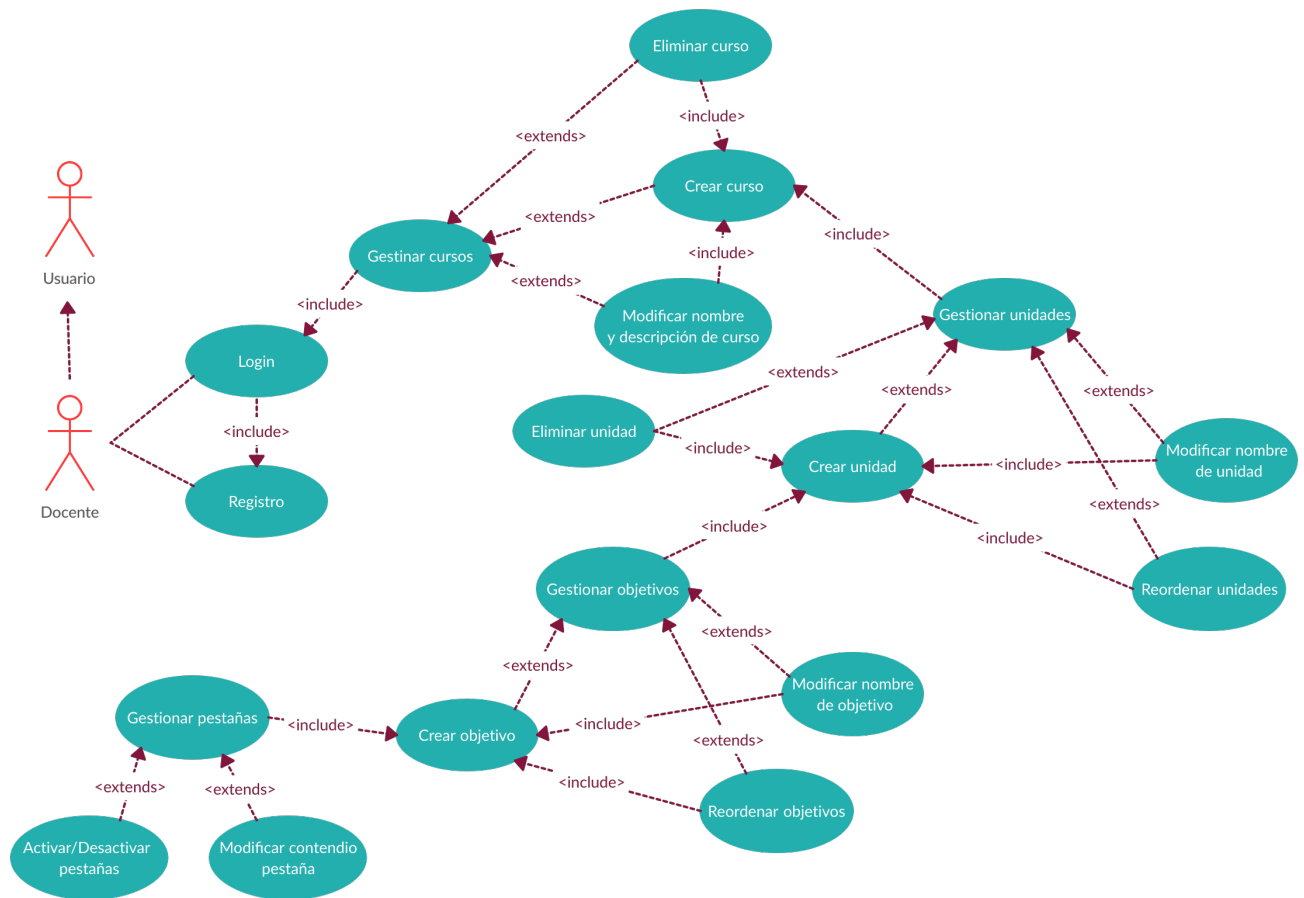


Figura B.1: Diagrama de casos de uso de gestión de contenido

B.2. Diagrama de caso de uso del banco de preguntas

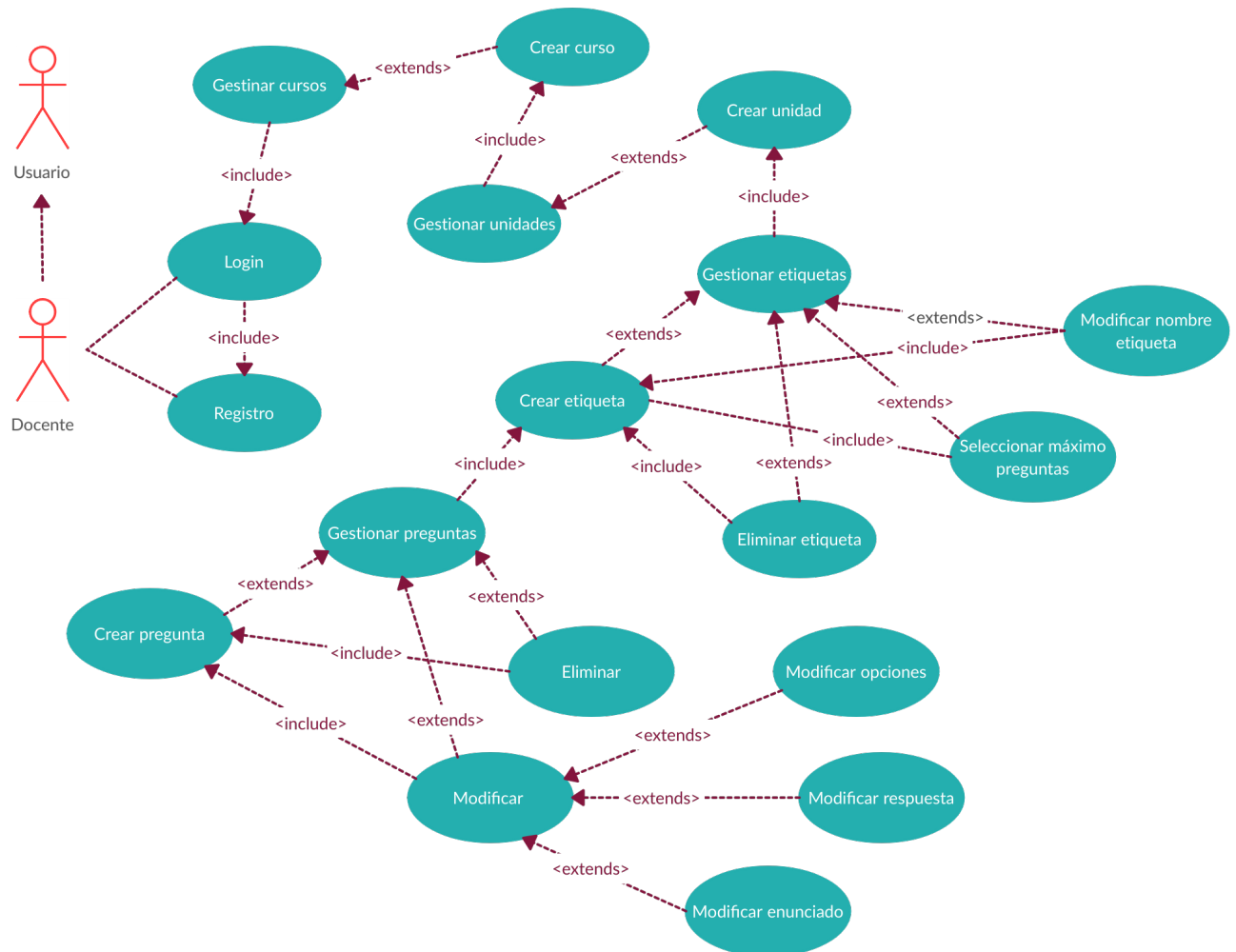


Figura B.2: Diagrama de casos de uso del banco de preguntas

B.3. Diagrama de caso de uso de la navegación de contenido

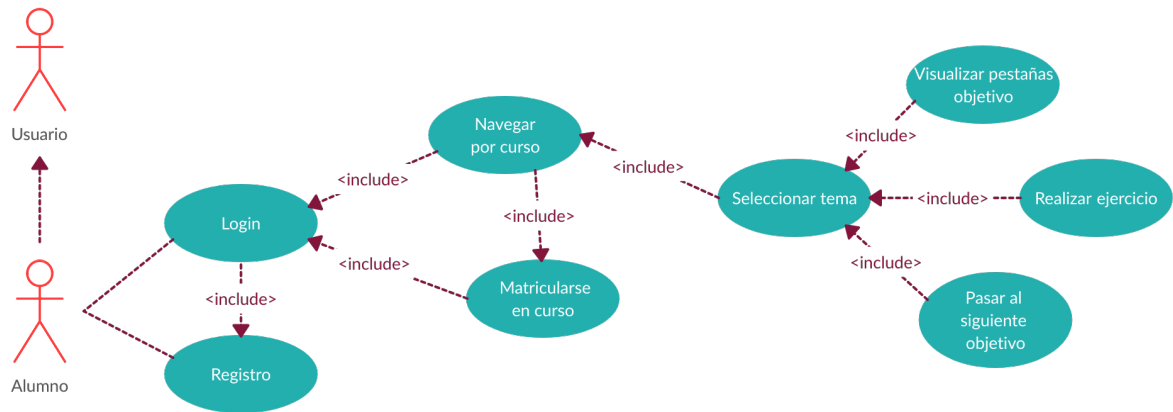


Figura B.3: Diagrama de casos de uso de la navegación de contenido

B.4. Diagrama de caso de uso del seguimiento

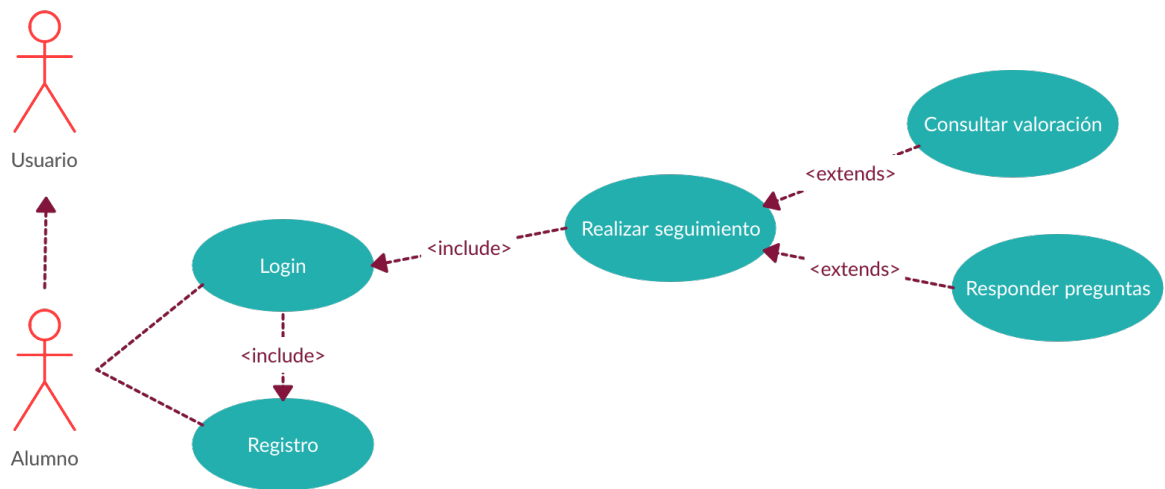


Figura B.4: Diagrama de casos de uso del seguimiento

C | Diagramas de flujo

C.1. Gestor de un curso

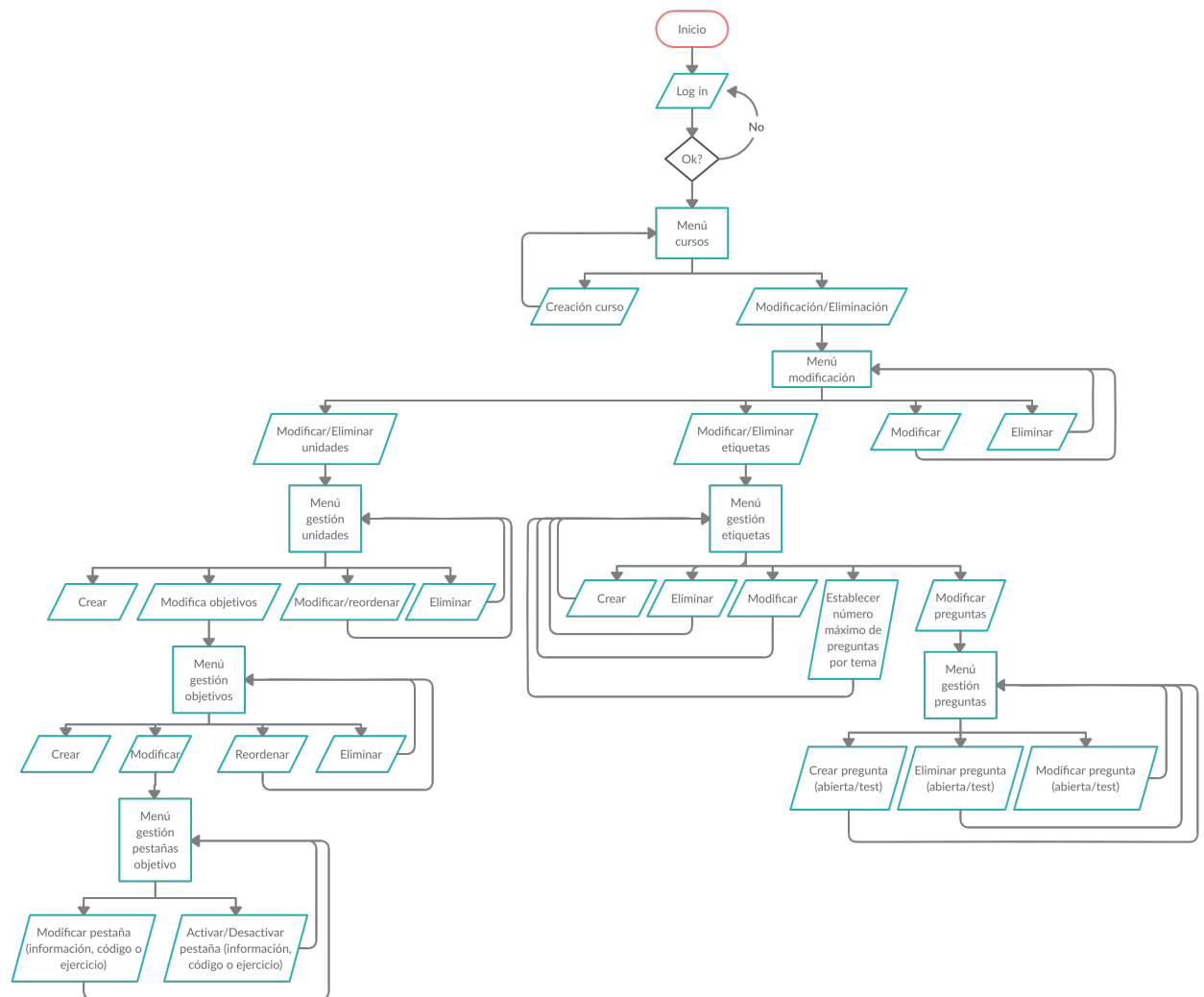


Figura C.1: Diagrama de flujo para el profesor.

C.2. Estudiante de un curso

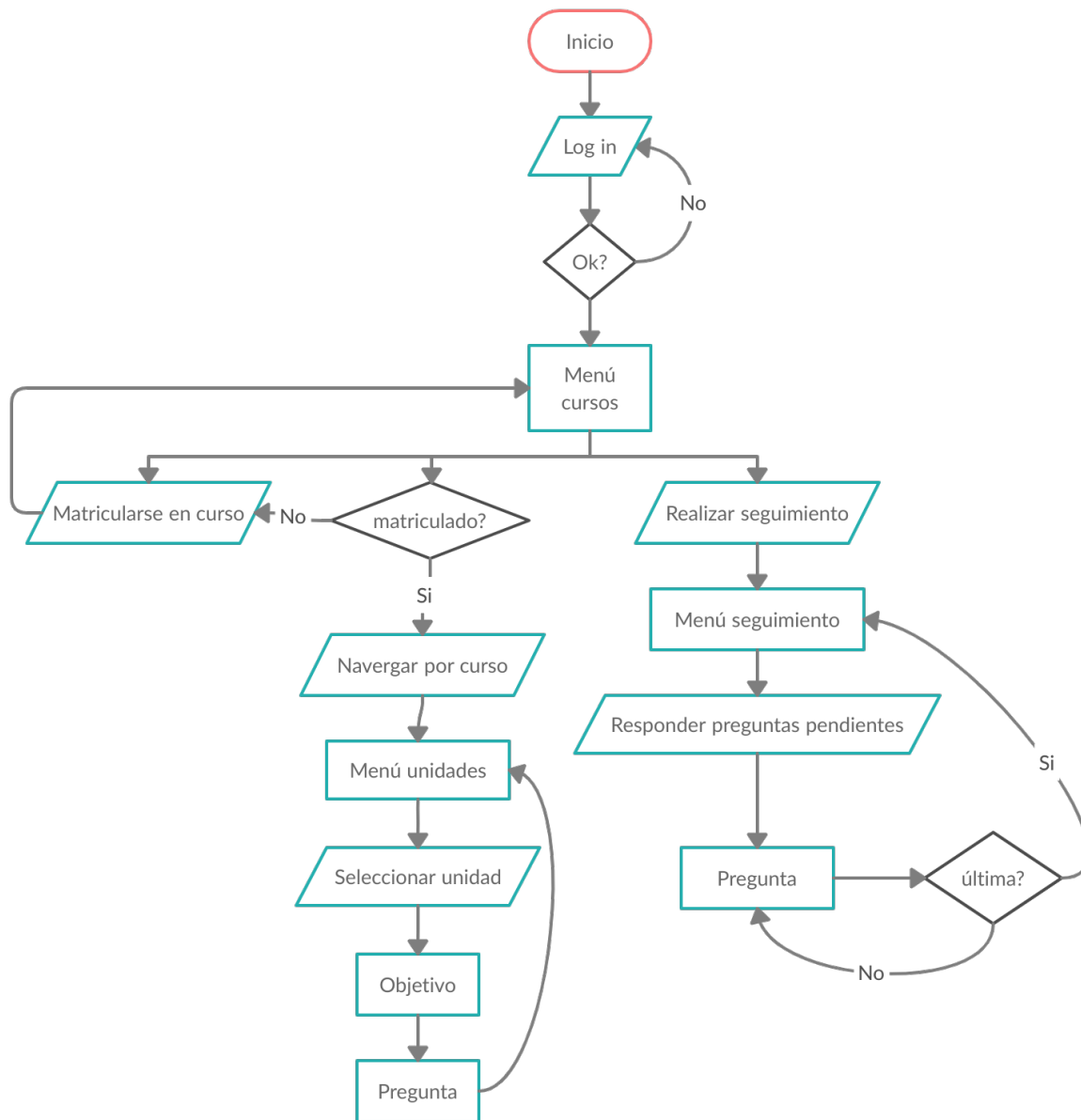


Figura C.2: Diagrama de flujo para el estudiante.

D | Mapa de navegación

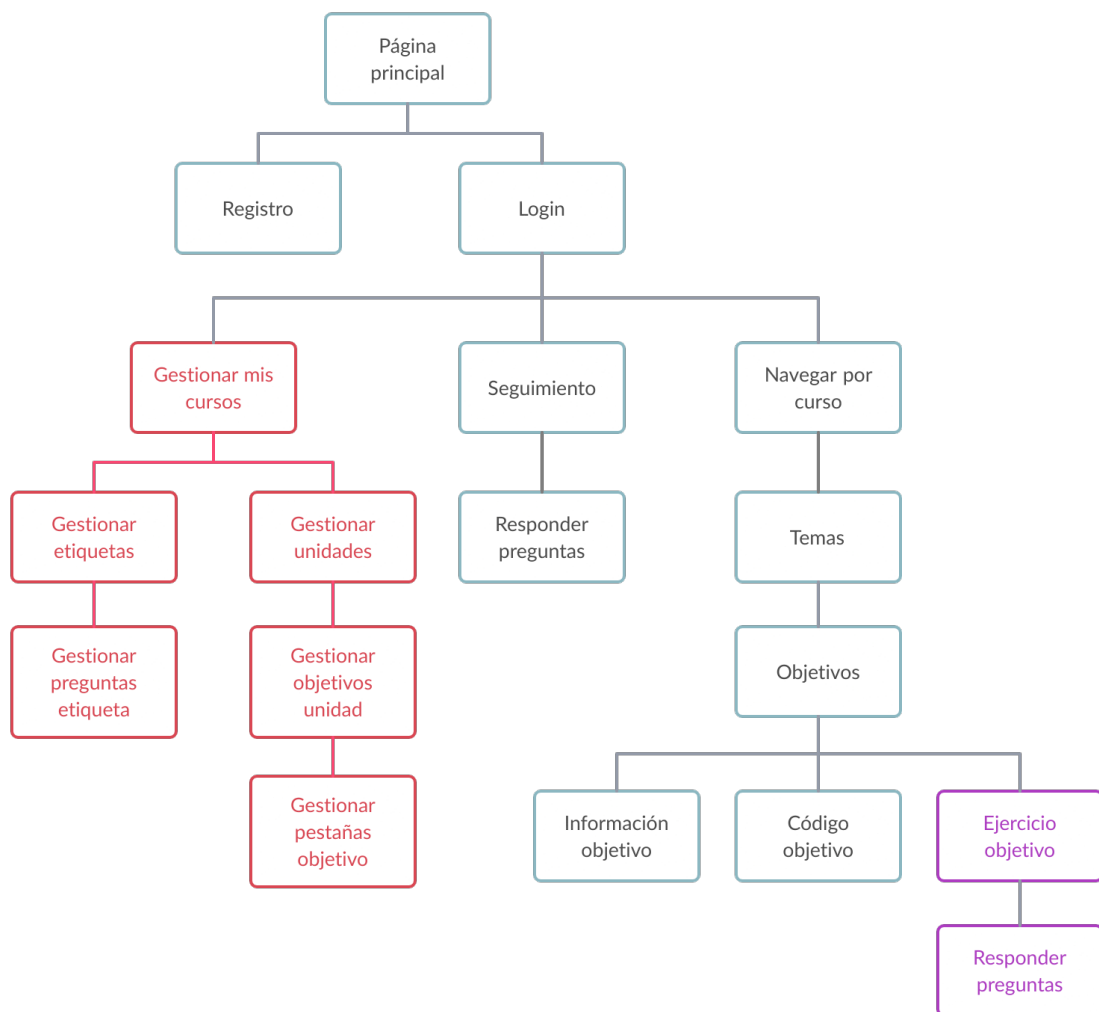


Figura D.1: Mapa de navegación de la herramienta Share&Learn.

E | Maquetas

E.1. Maquetas Web

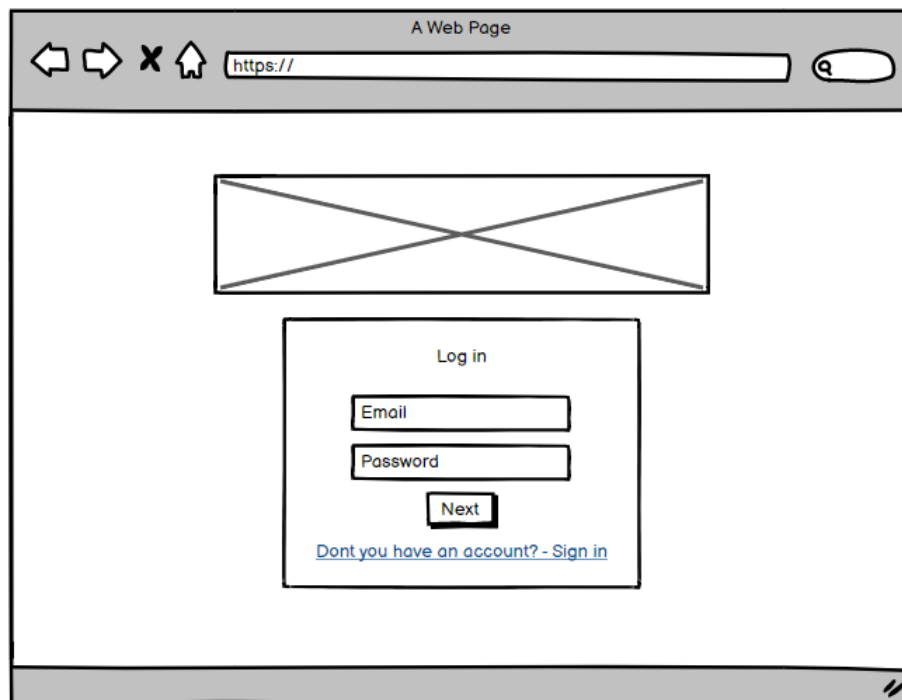


Figura E.1: Maqueta de vista de inicio de sesión Web.

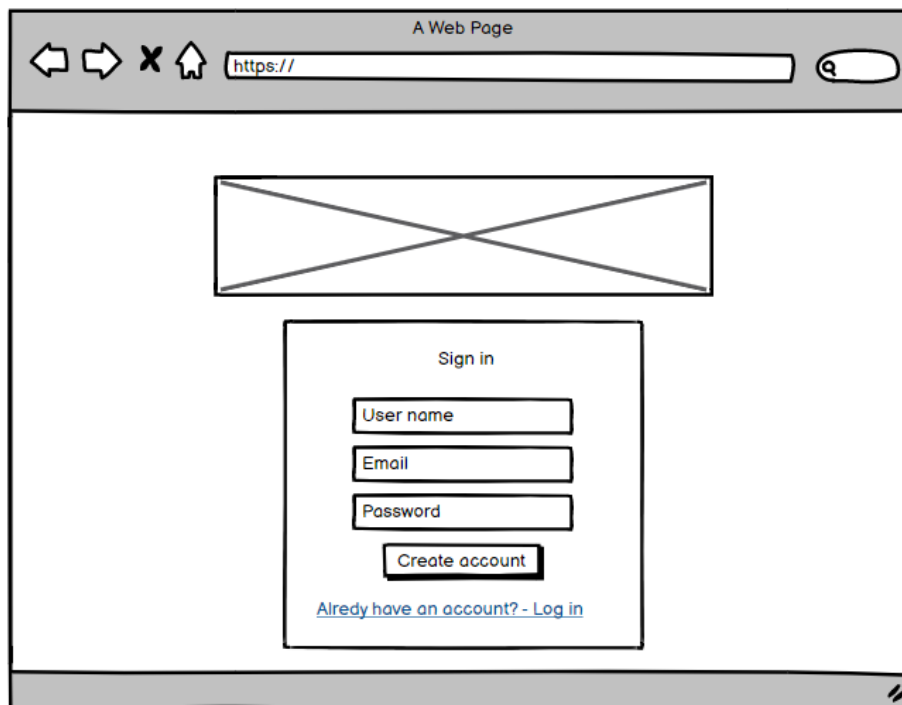


Figura E.2: Maqueta de vista de registro de usuario Web.

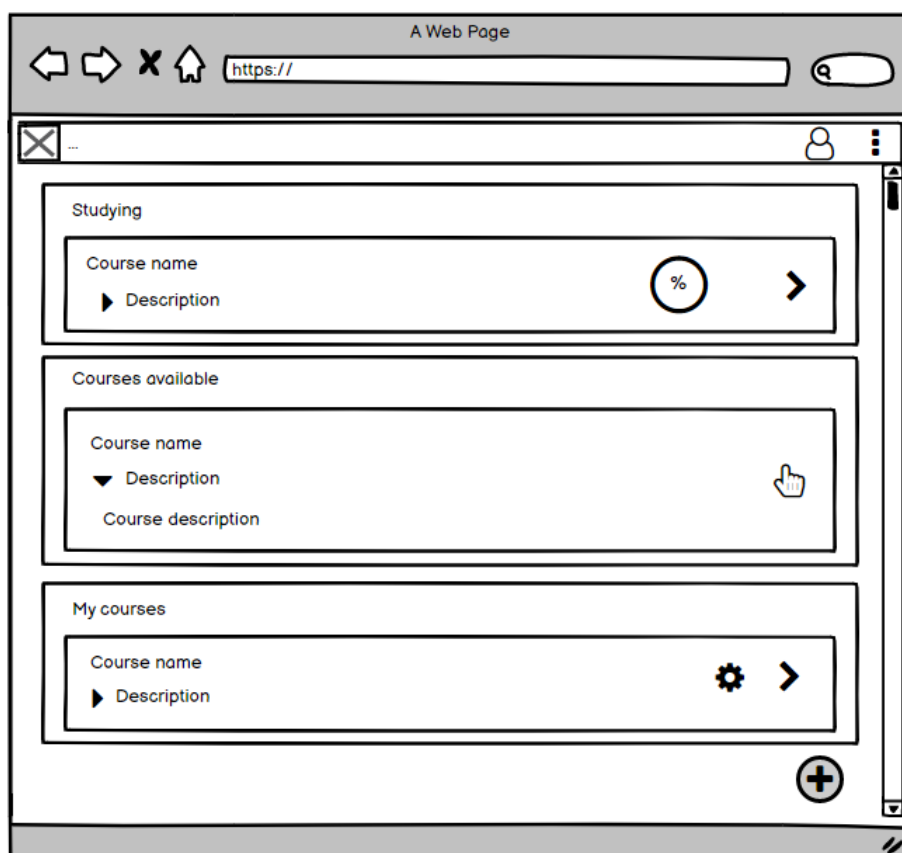


Figura E.3: Maqueta de vista de la pantalla principal Web.

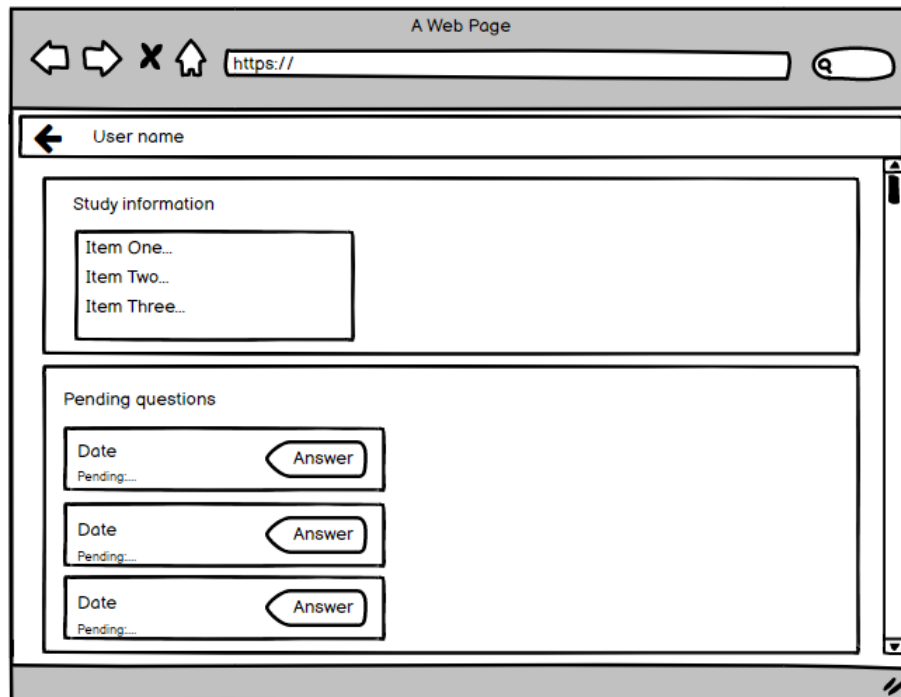


Figura E.4: Maqueta de vista del seguimiento de usuario Web.

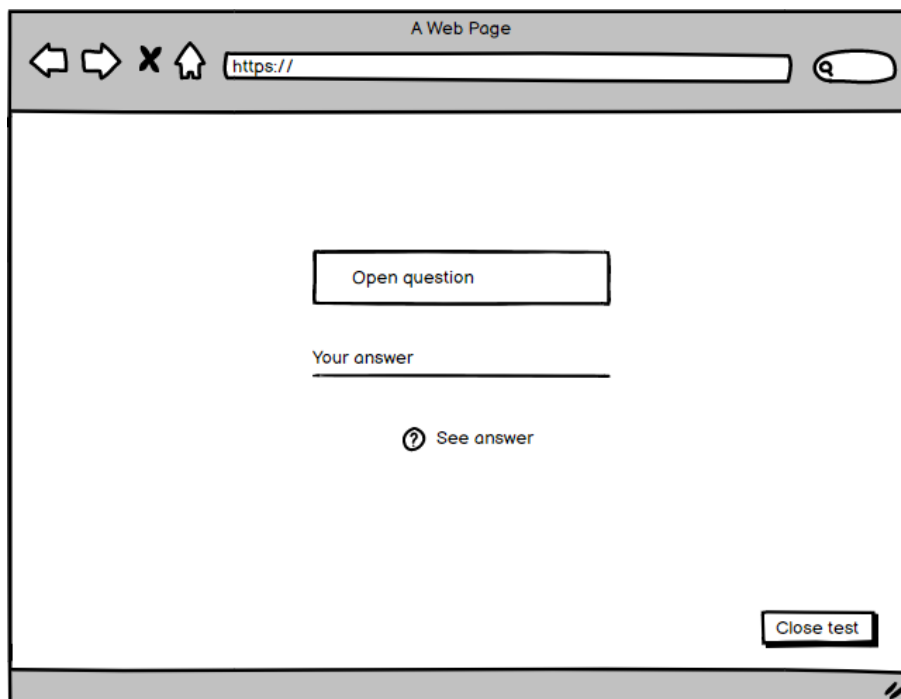


Figura E.5: Maqueta de vista de pregunta abierta Web.

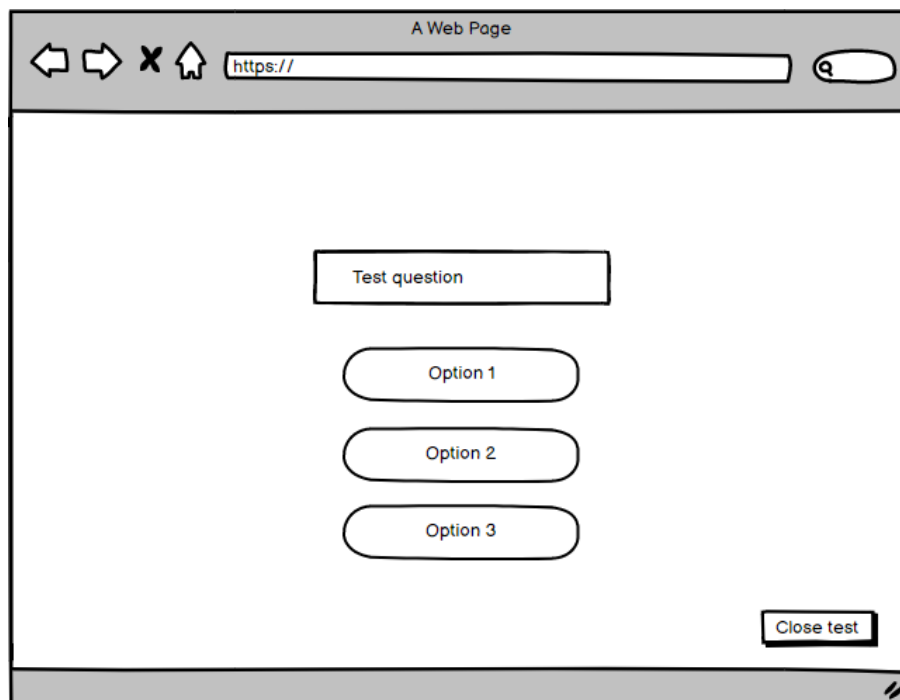


Figura E.6: Maqueta de vista de pregunta tipo test Web.

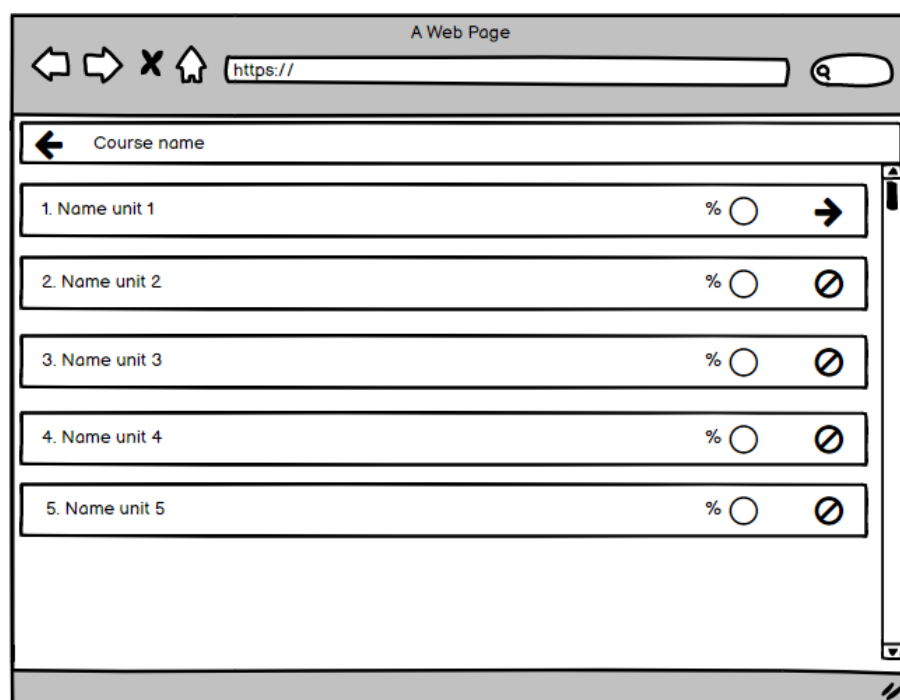


Figura E.7: Maqueta de vista de lista de unidades Web.

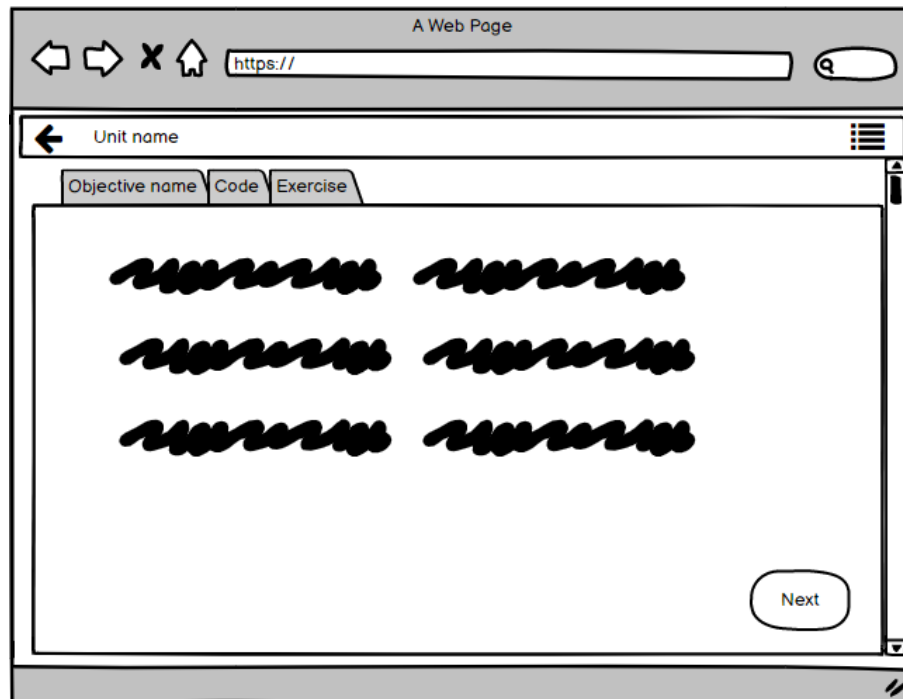


Figura E.8: Maqueta de vista de objetivos Web.

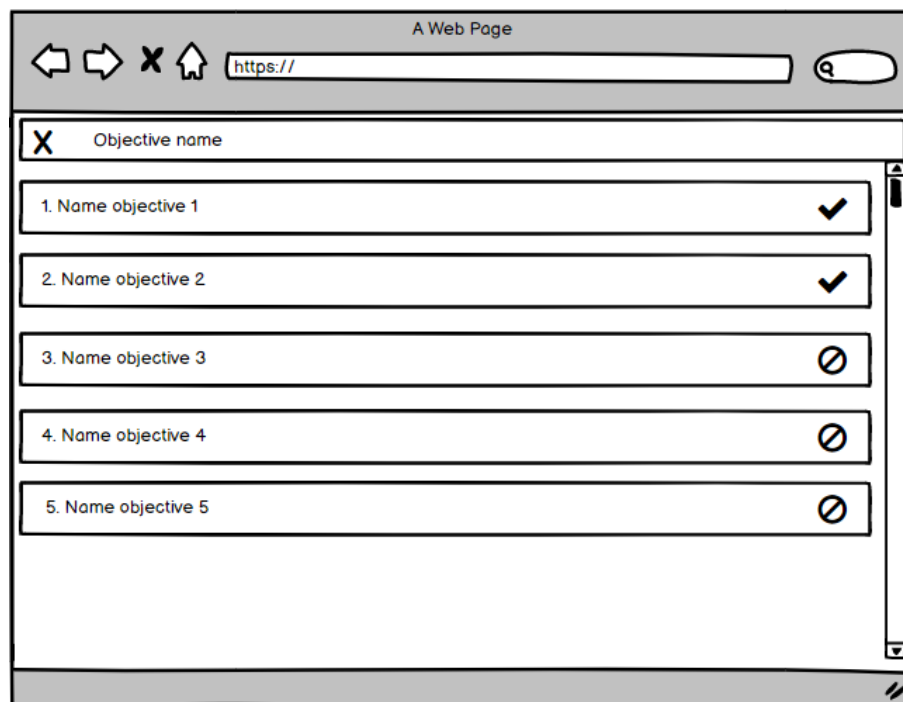


Figura E.9: Maqueta de vista de lista de objetivos Web.

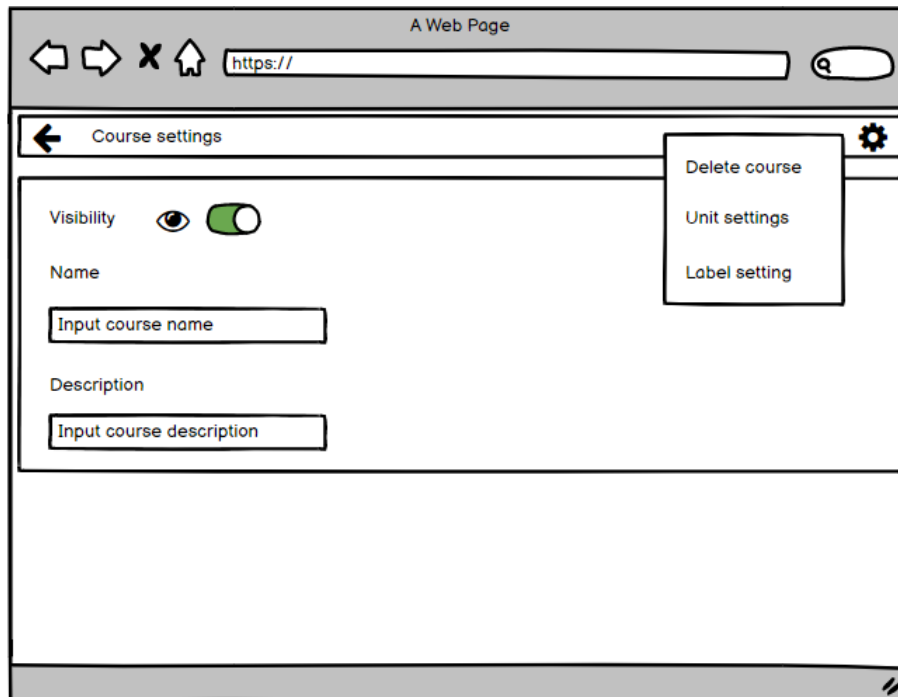


Figura E.10: Maqueta de vista de ajuste de cursos Web.

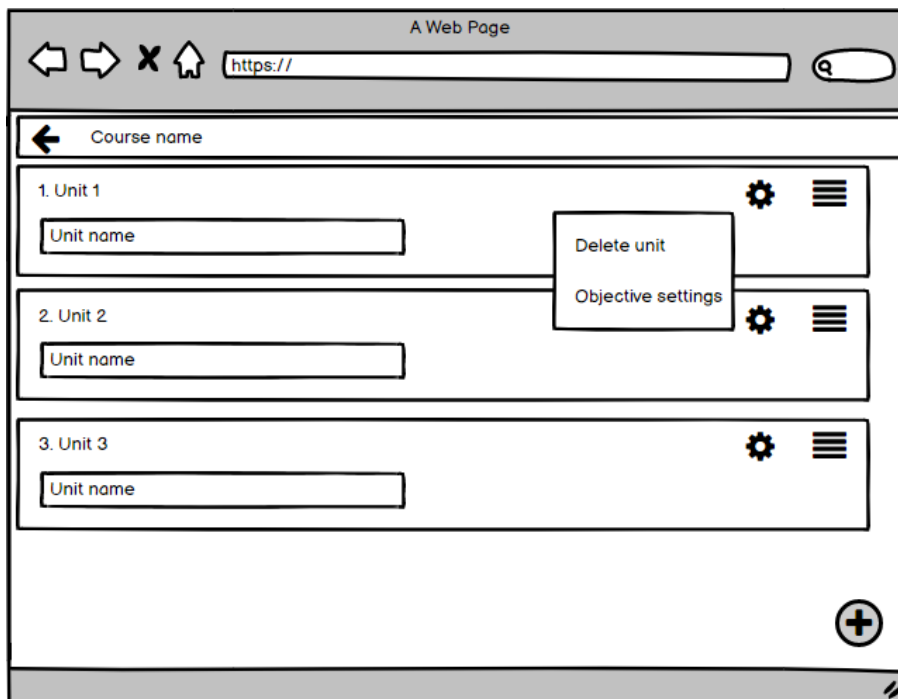


Figura E.11: Maqueta de vista de ajuste de unidades Web.

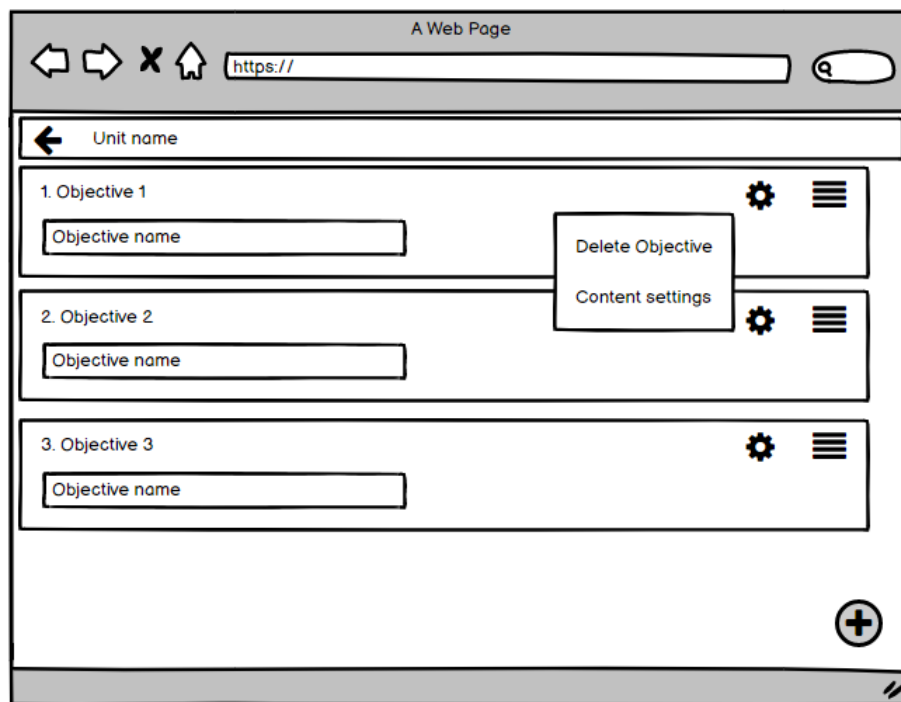


Figura E.12: Maqueta de vista de ajuste de objetivos Web.

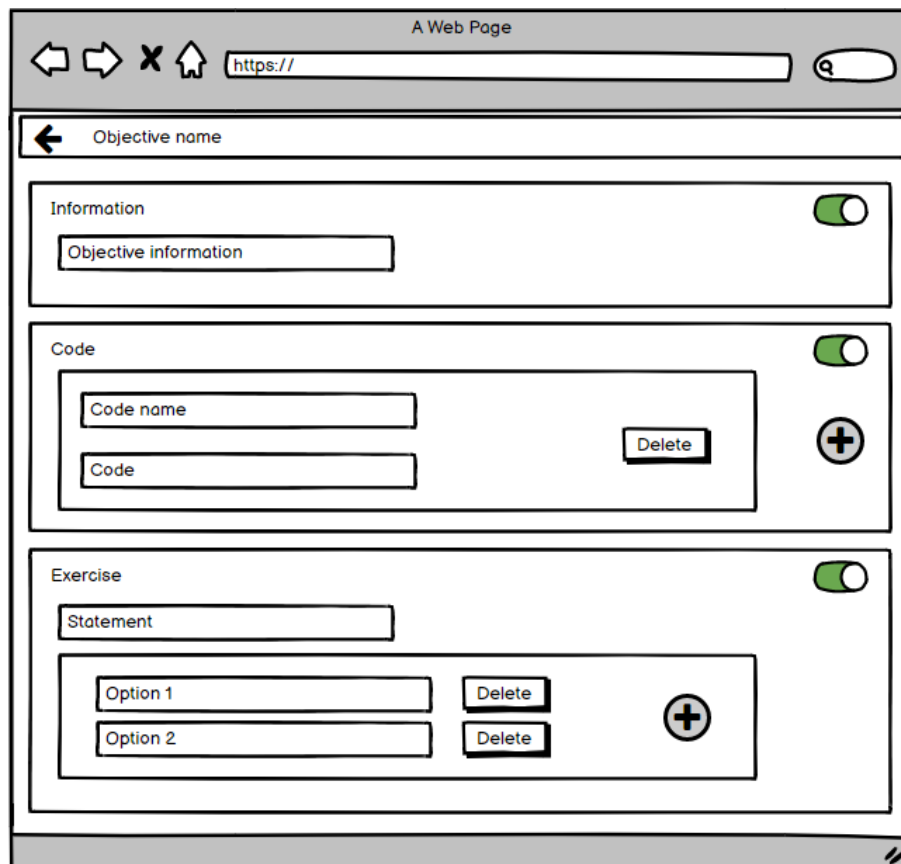


Figura E.13: Maqueta de vista de ajuste de contenido Web.

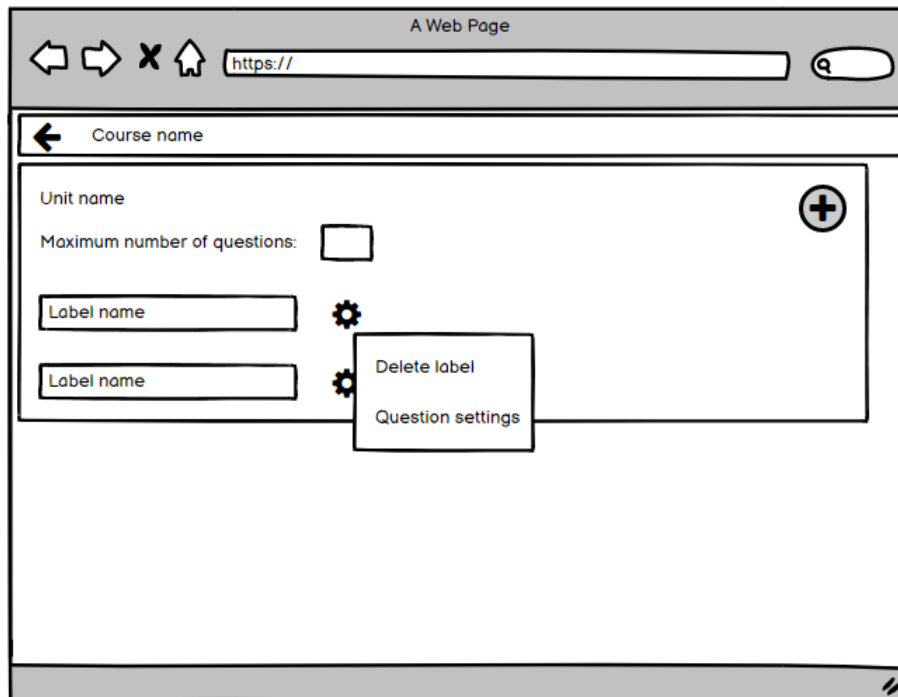


Figura E.14: Maqueta de vista de ajuste de etiquetas Web.

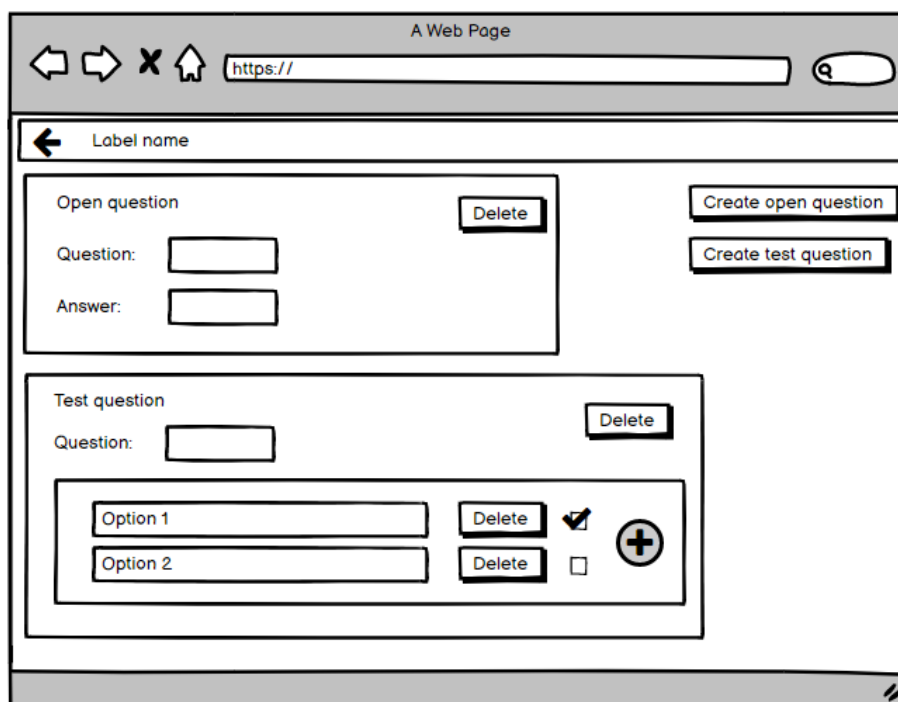


Figura E.15: Maqueta de vista de ajuste de preguntas Web.

E.2. Maquetas móvil

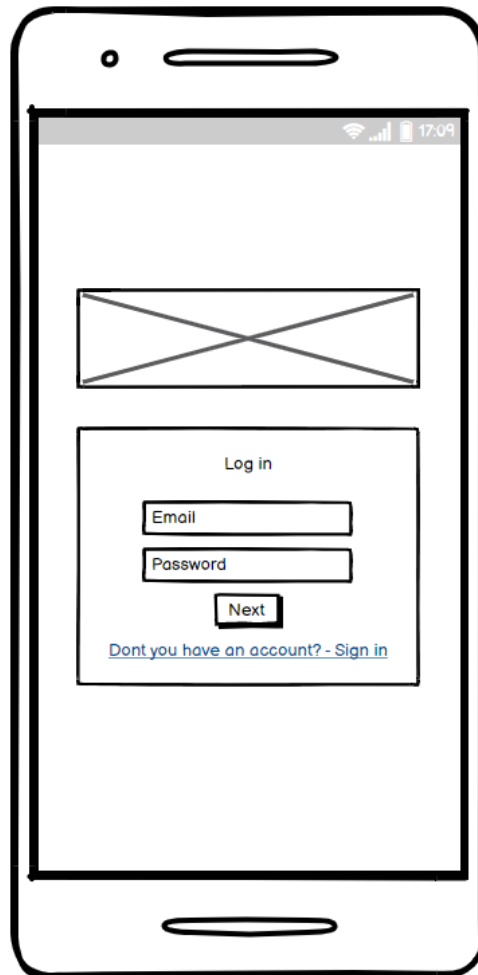


Figura E.16: Maqueta de vista de inicio de sesión móvil.

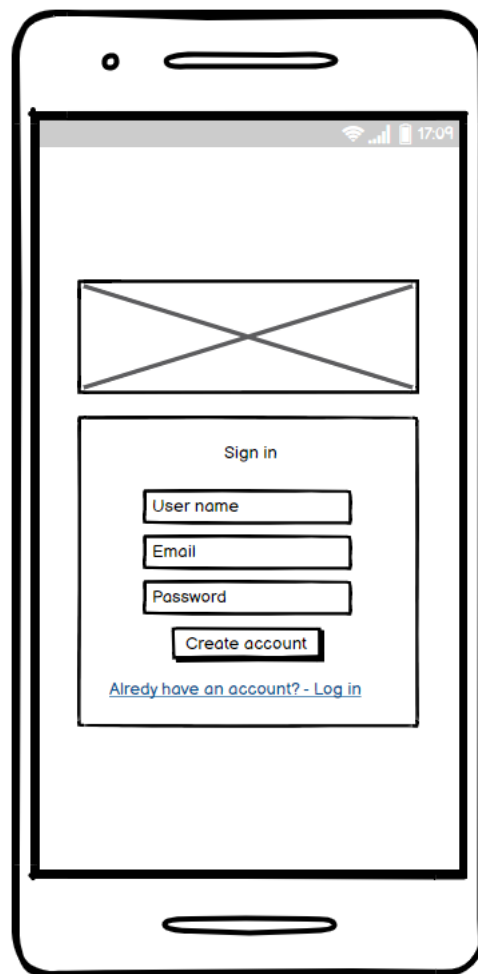


Figura E.17: Maqueta de vista de registro de usuario móvil.

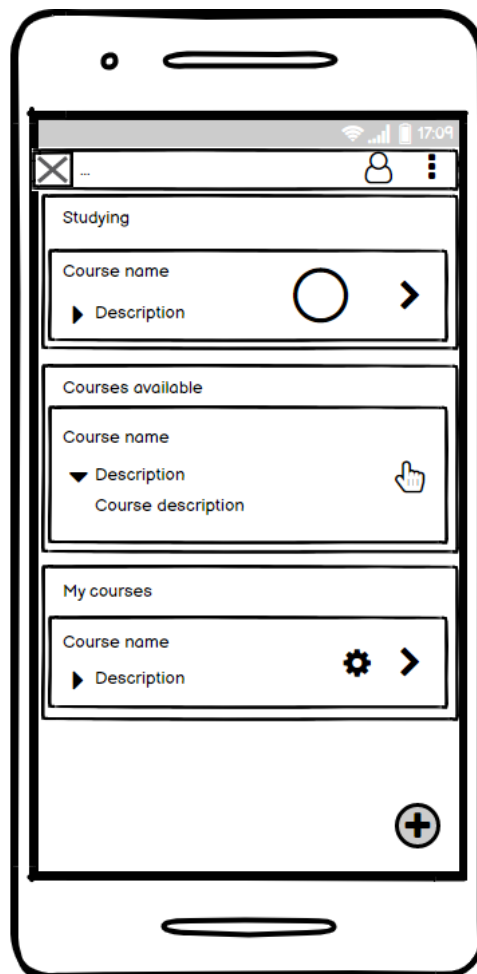


Figura E.18: Maqueta de vista de la pantalla principal móvil.

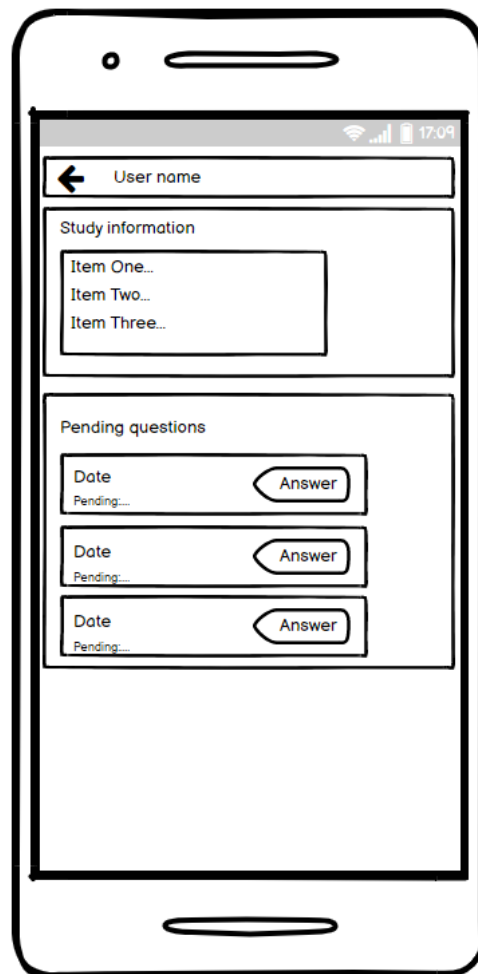


Figura E.19: Maqueta de vista del seguimiento de usuario móvil.

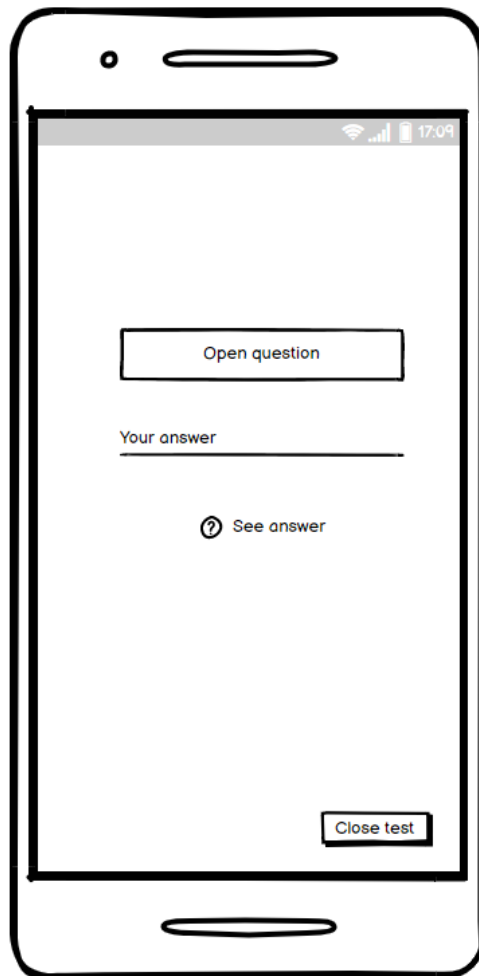


Figura E.20: Maqueta de vista de pregunta abierta móvil.

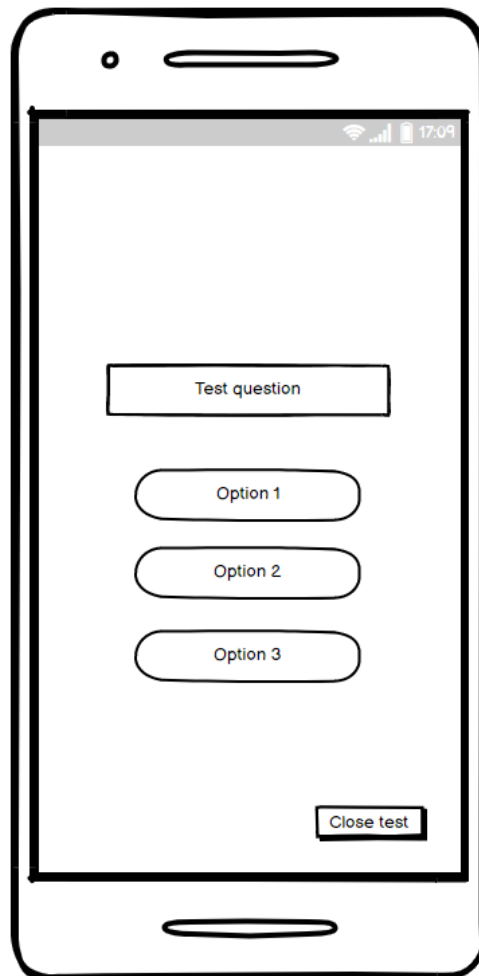


Figura E.21: Maqueta de vista de pregunta tipo test móvil.

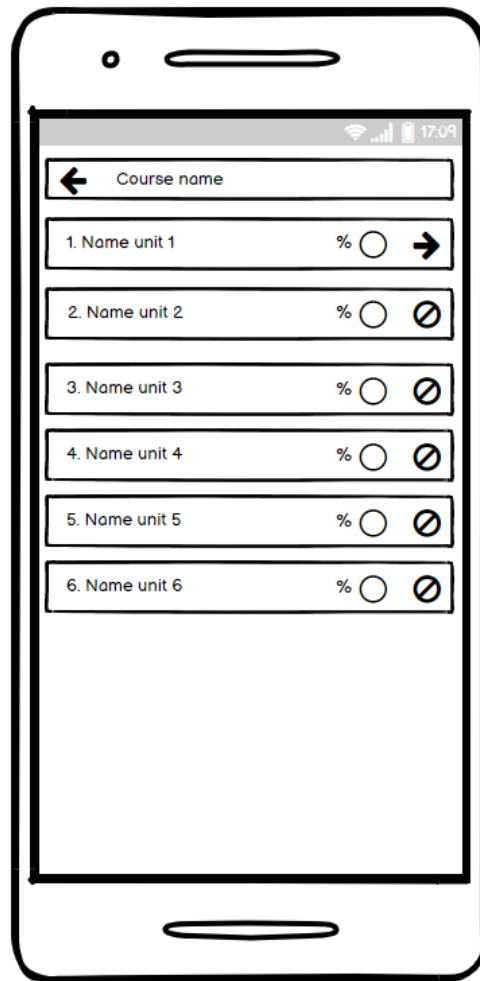


Figura E.22: Maqueta de vista de lista de unidades móvil.

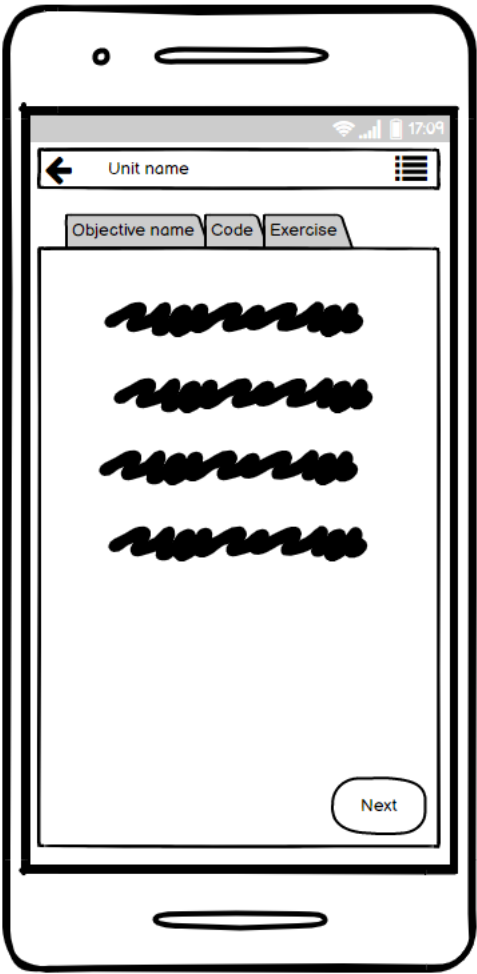


Figura E.23: Maqueta de vista de objetivos móvil.

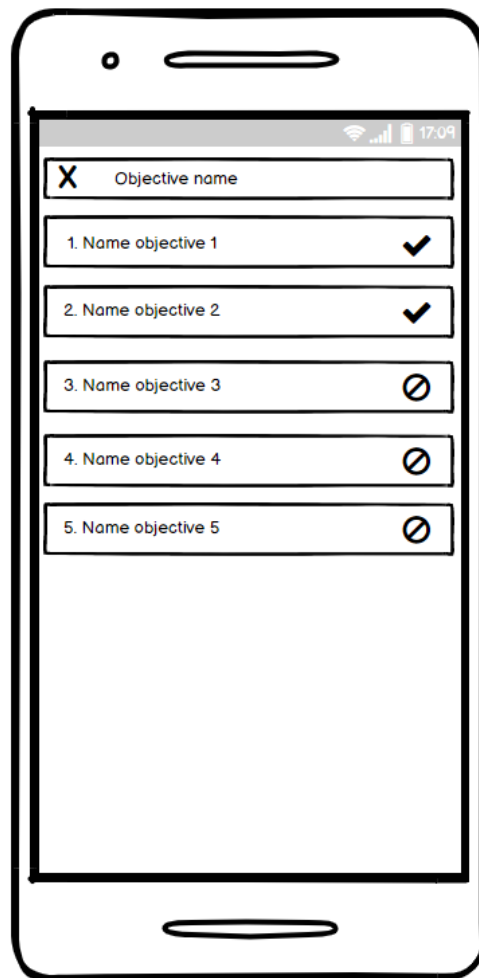


Figura E.24: Maqueta de vista de lista de objetivos móvil.

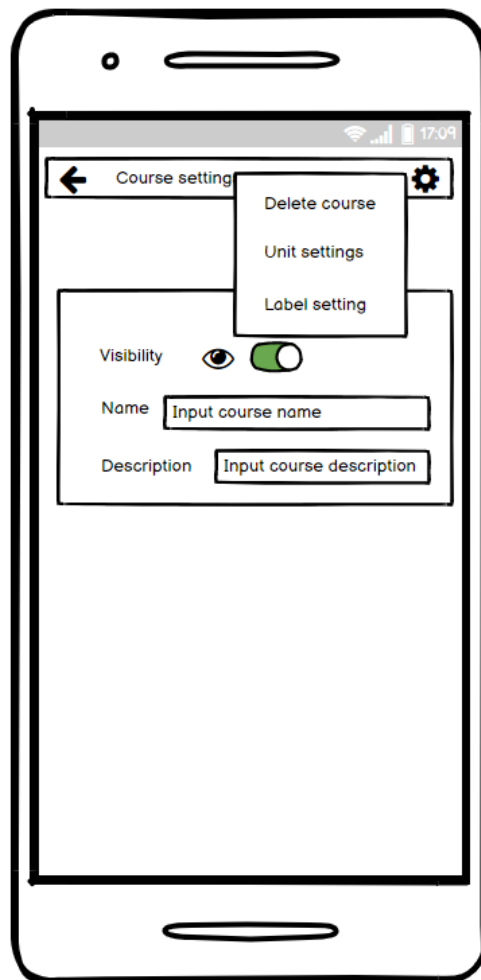


Figura E.25: Maqueta de vista de ajuste de cursos móvil.

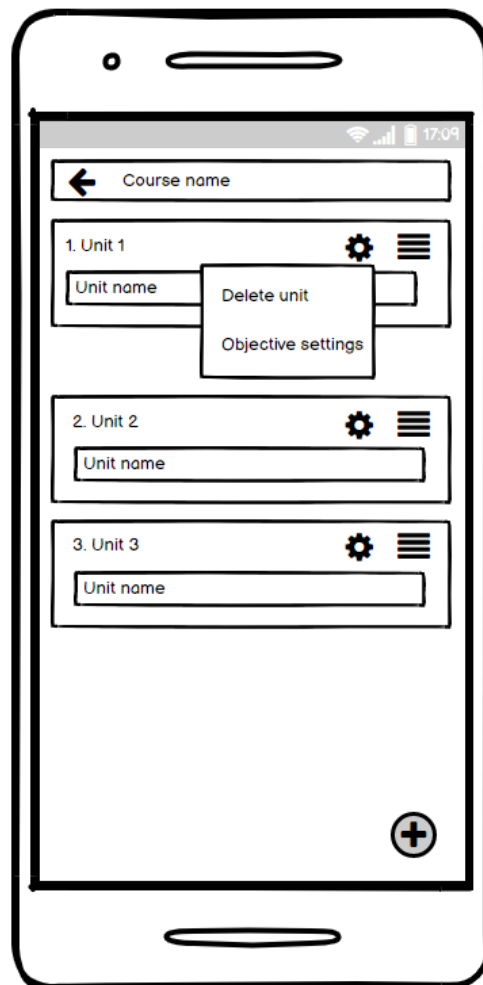


Figura E.26: Maqueta de vista de ajuste de unidades móvil.

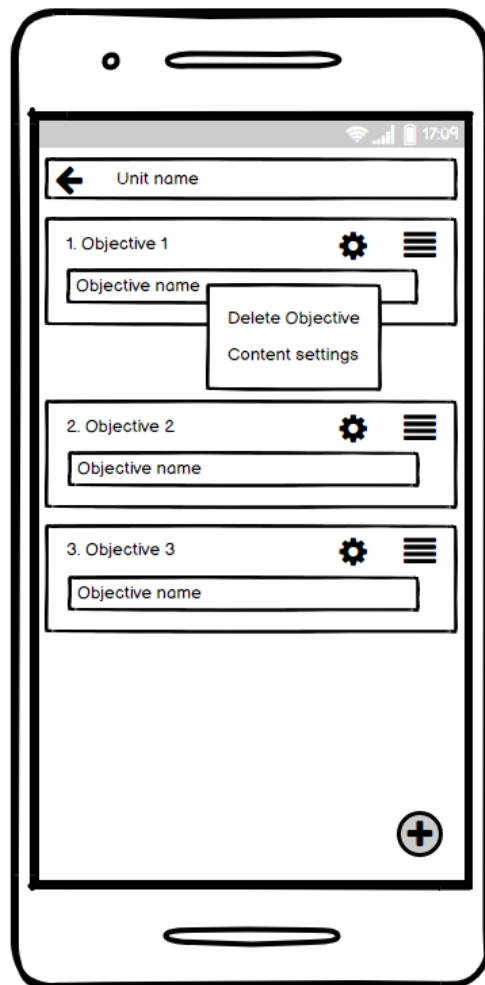


Figura E.27: Maqueta de vista de ajuste de objetivos móvil.

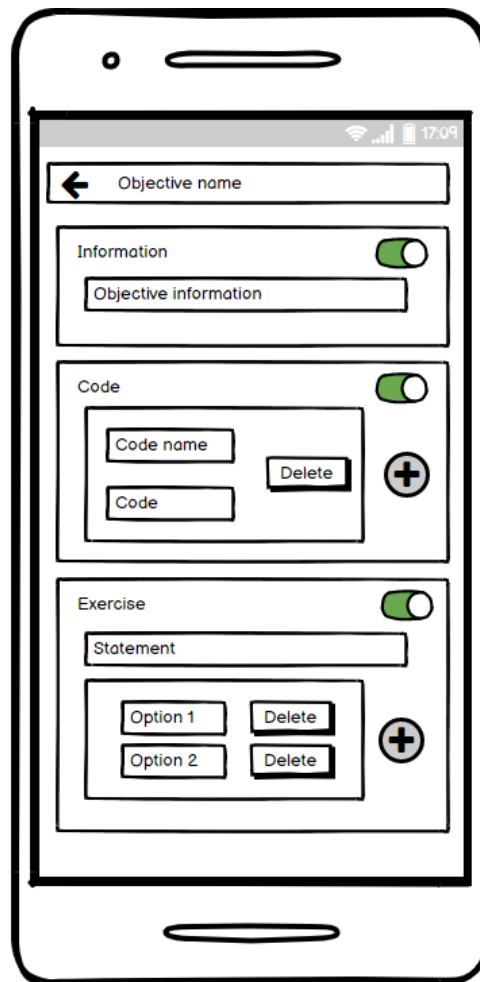


Figura E.28: Maqueta de vista de ajuste de contenido móvil.

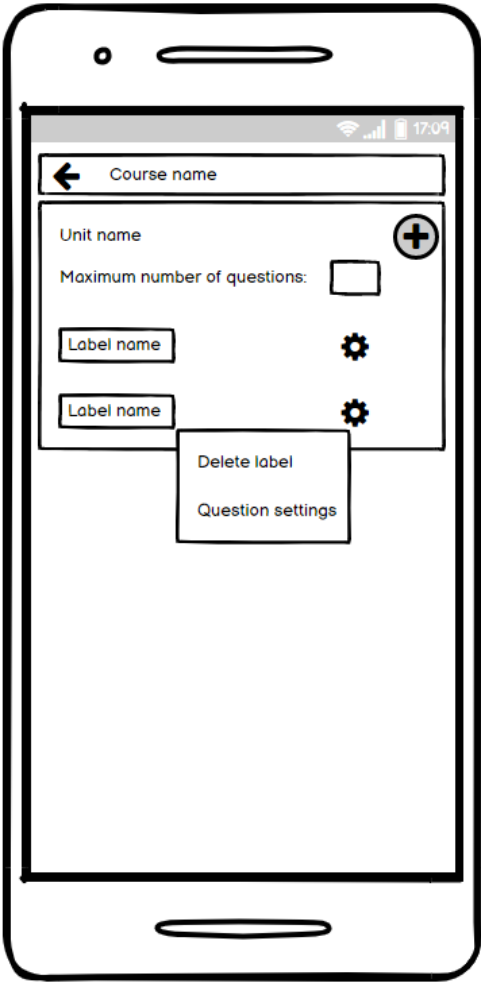


Figura E.29: Maqueta de vista de ajuste de etiquetas móvil.

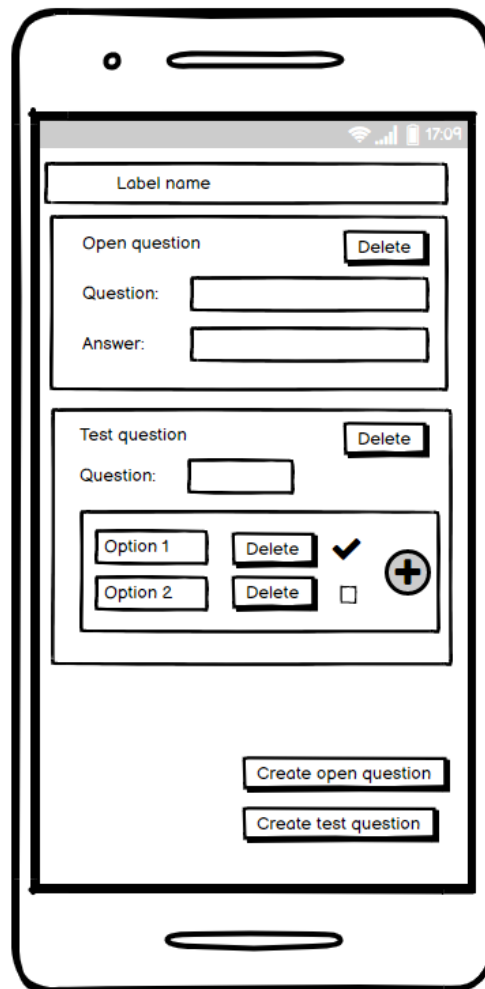


Figura E.30: Maqueta de vista de ajuste de preguntas móvil.

F | Producto final

F.1. Vista Web

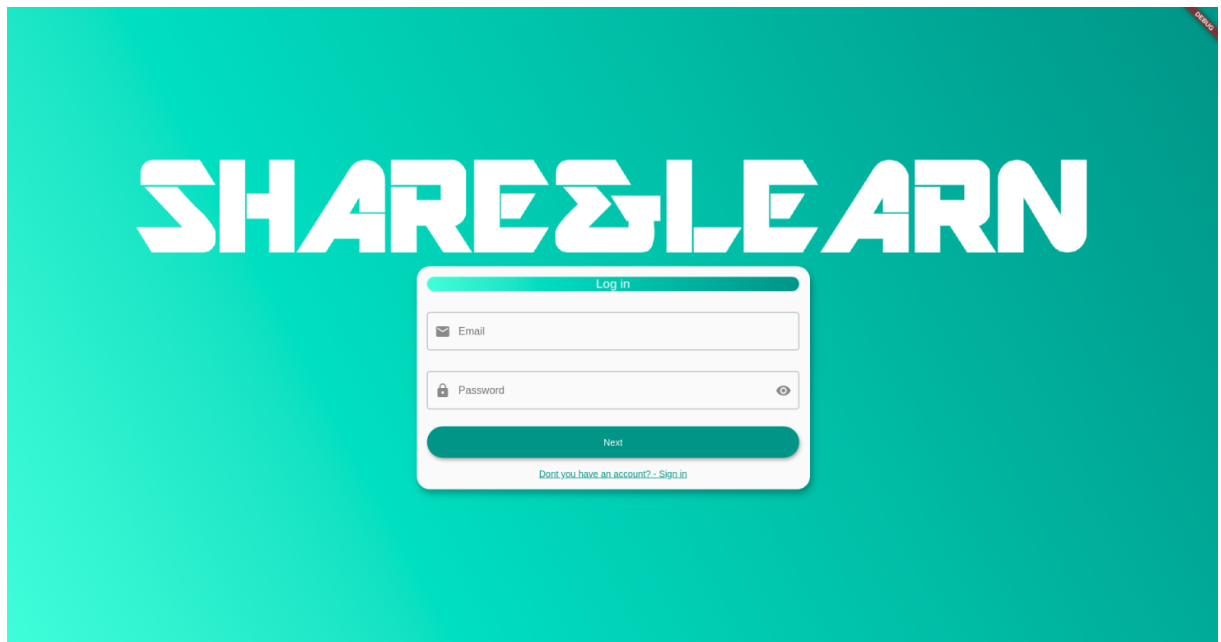


Figura F.1: Vista de inicio de sesión Web.

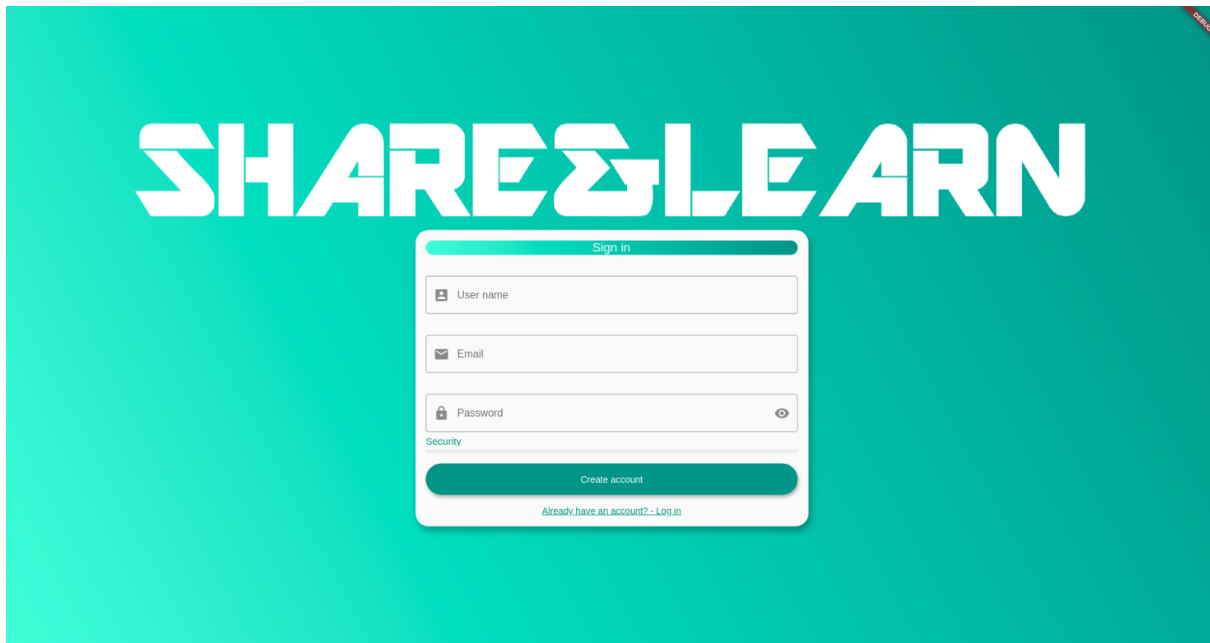


Figura F.2: Vista de registro de usuario Web.

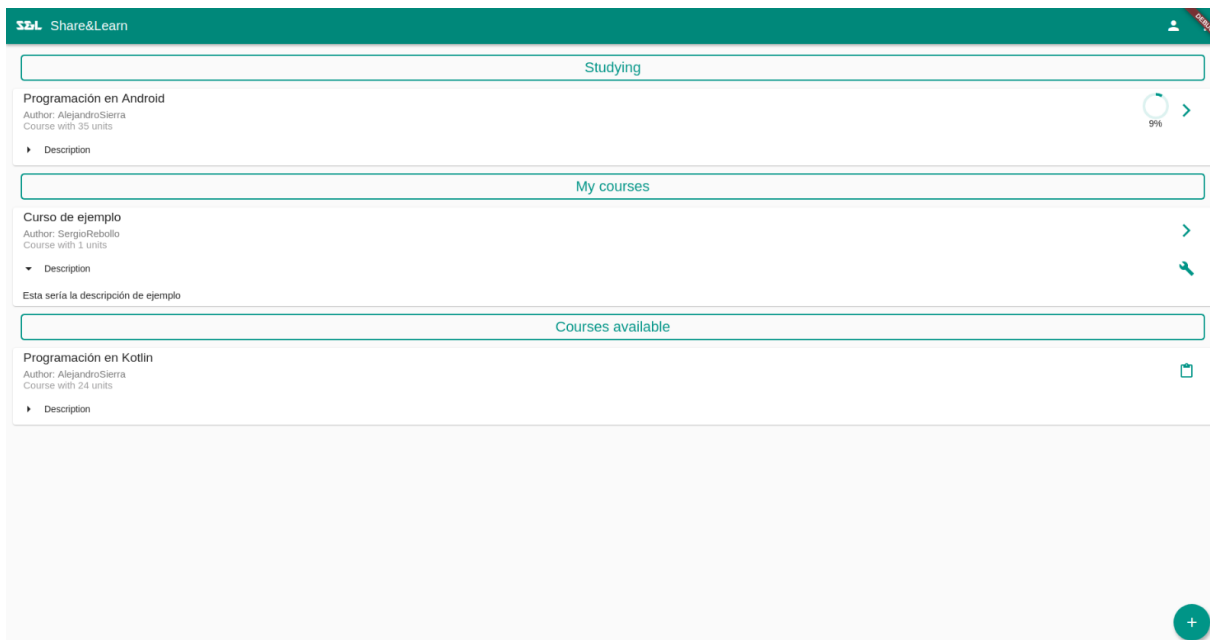


Figura F.3: Vista de pantalla principal Web.

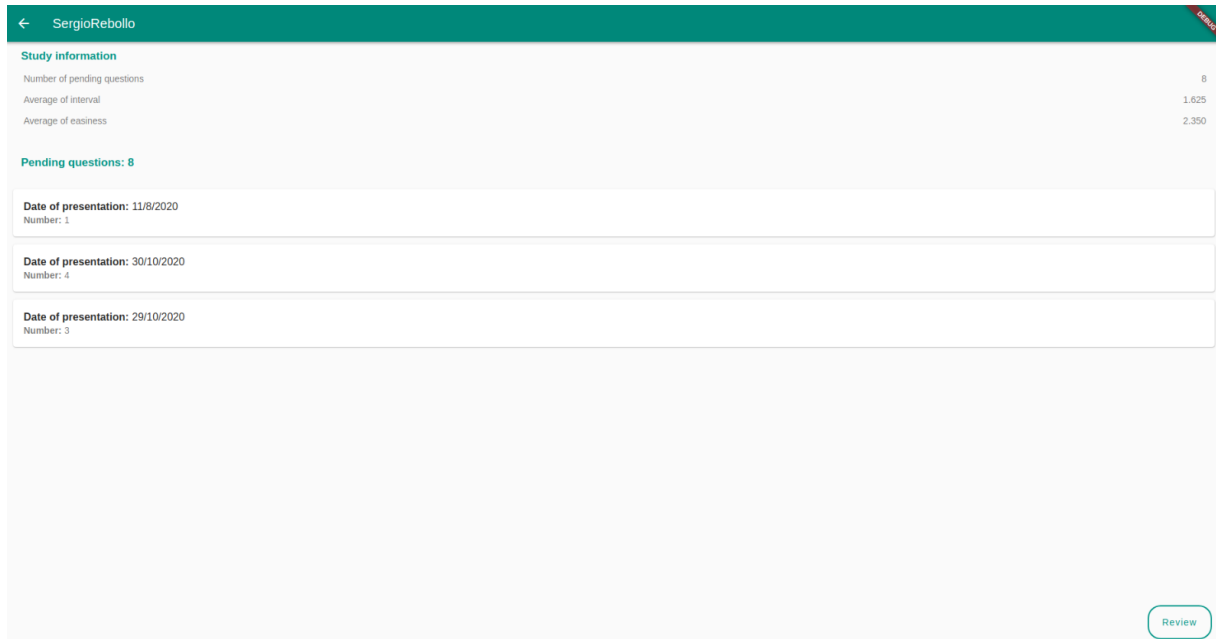


Figura F.4: Vista de seguimiento Web.

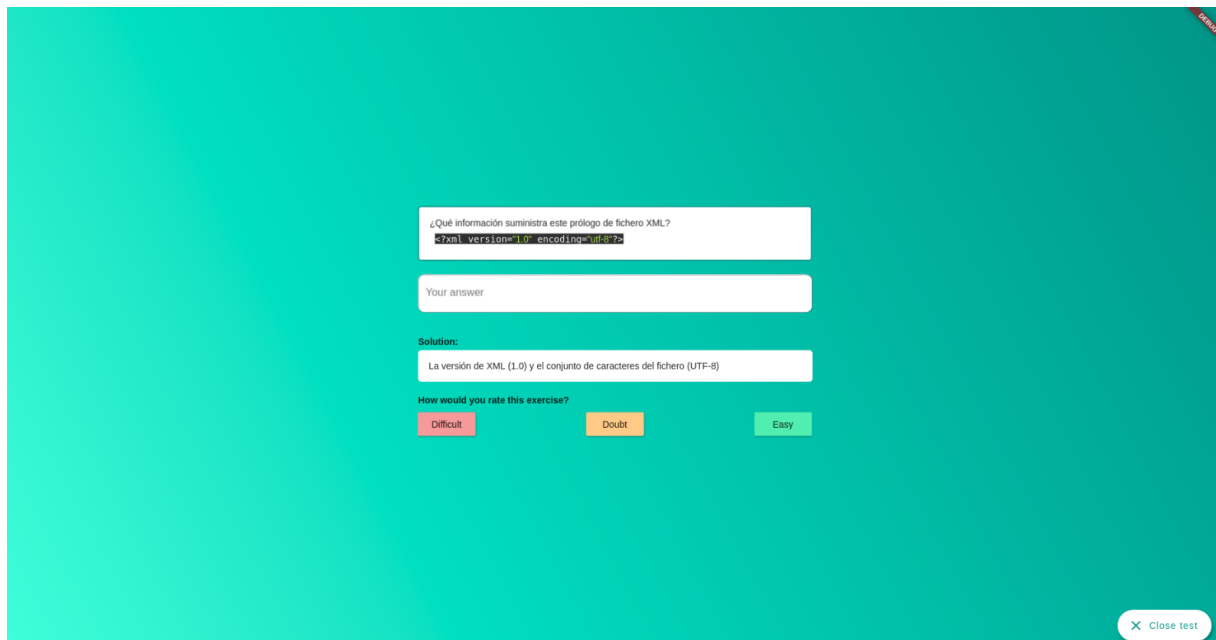


Figura F.5: Vista de pregunta abierta Web.

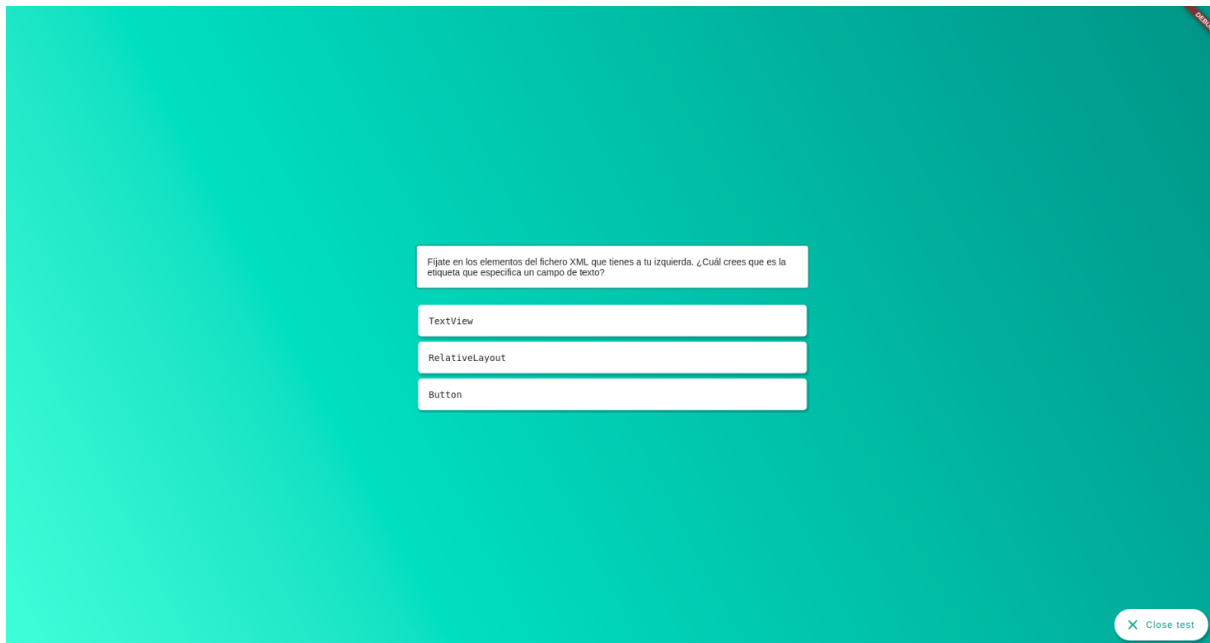


Figura F.6: Vista de pregunta tipo test Web.



Figura F.7: Vista de lista de unidades Web.

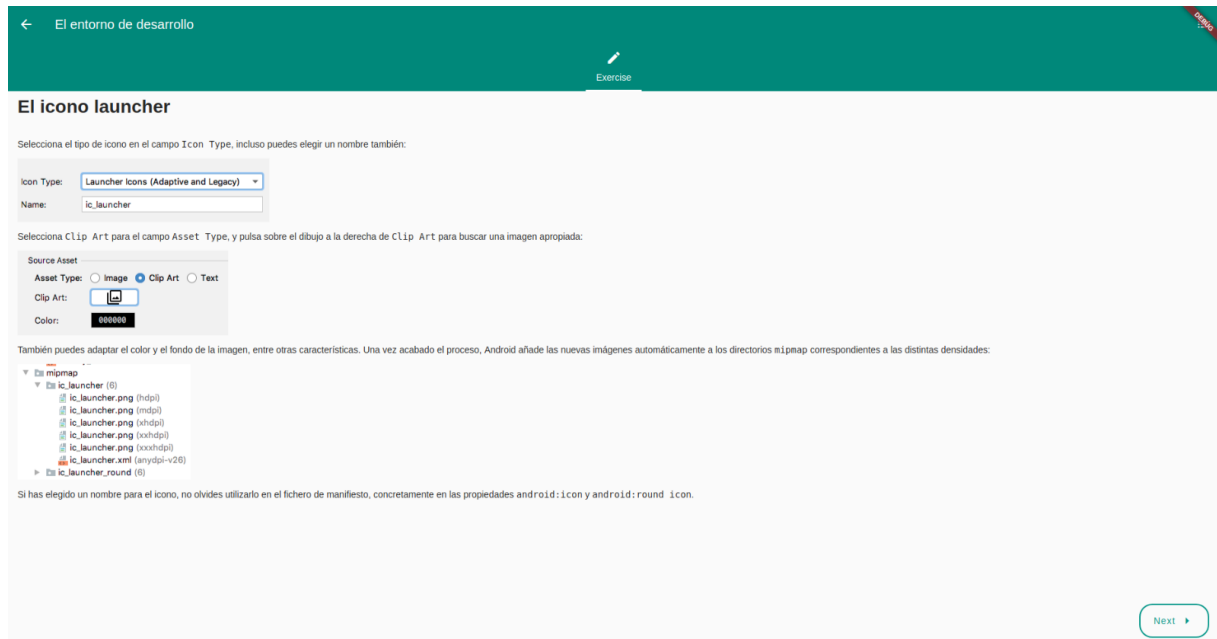


Figura F.8: Vista de objetivo 1 Web.

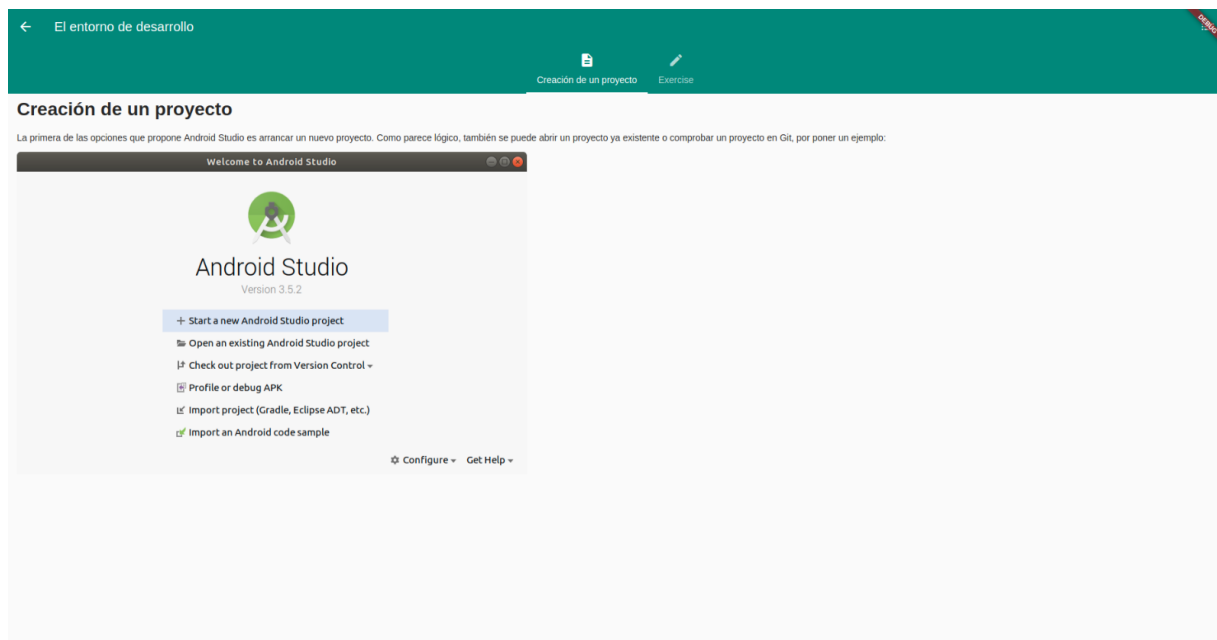


Figura F.9: Vista de objetivo 2 Web.

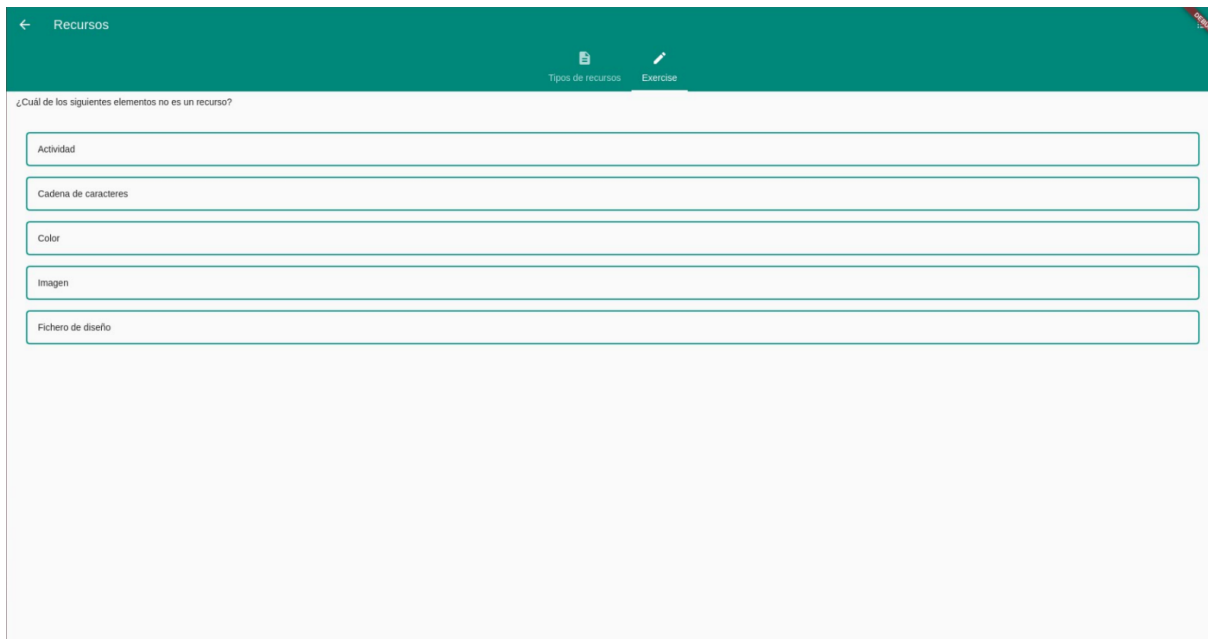


Figura F.10: Vista de objetivo 3 Web.

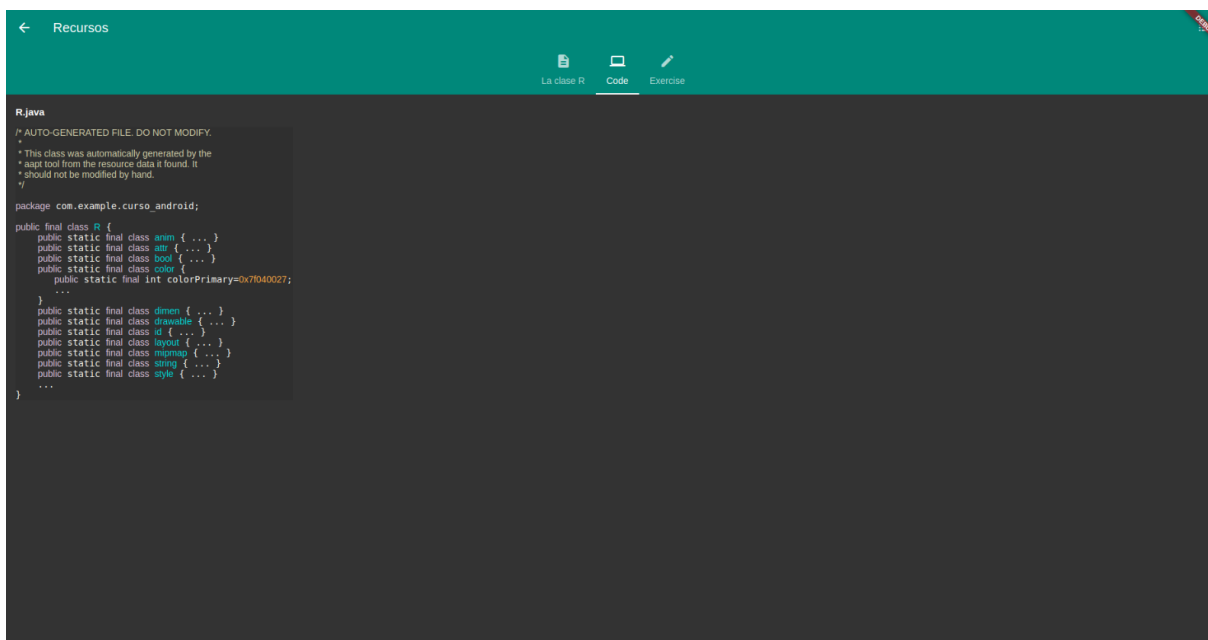


Figura F.11: Vista de objetivo 4 Web.

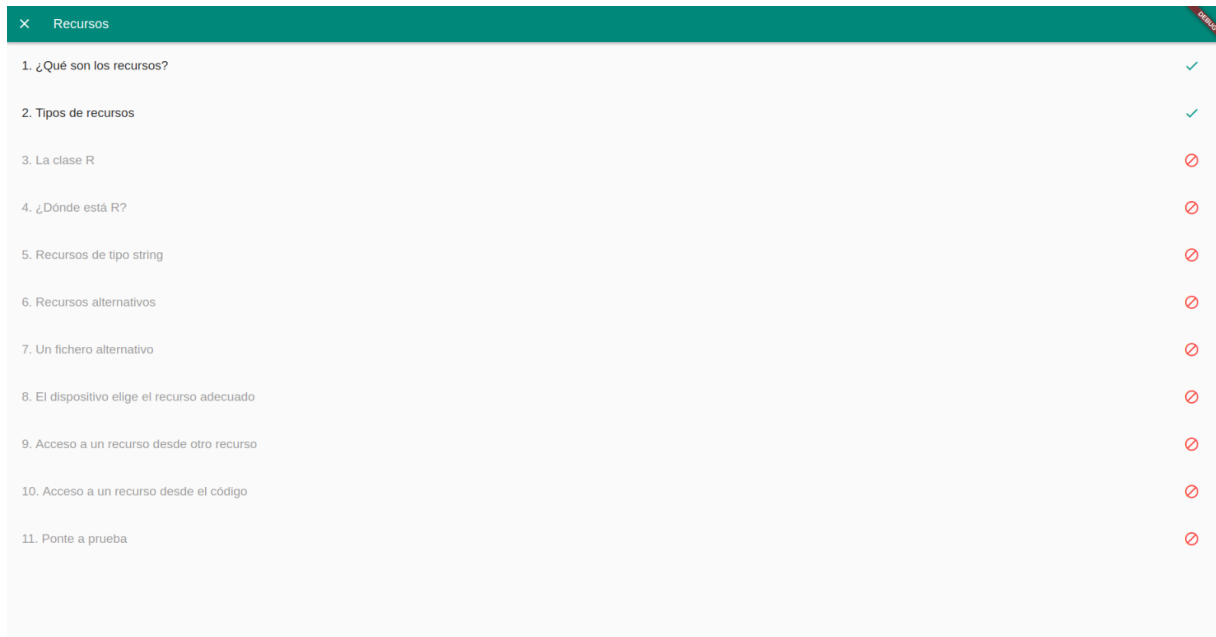


Figura F.12: Vista de lista de objetivos Web.

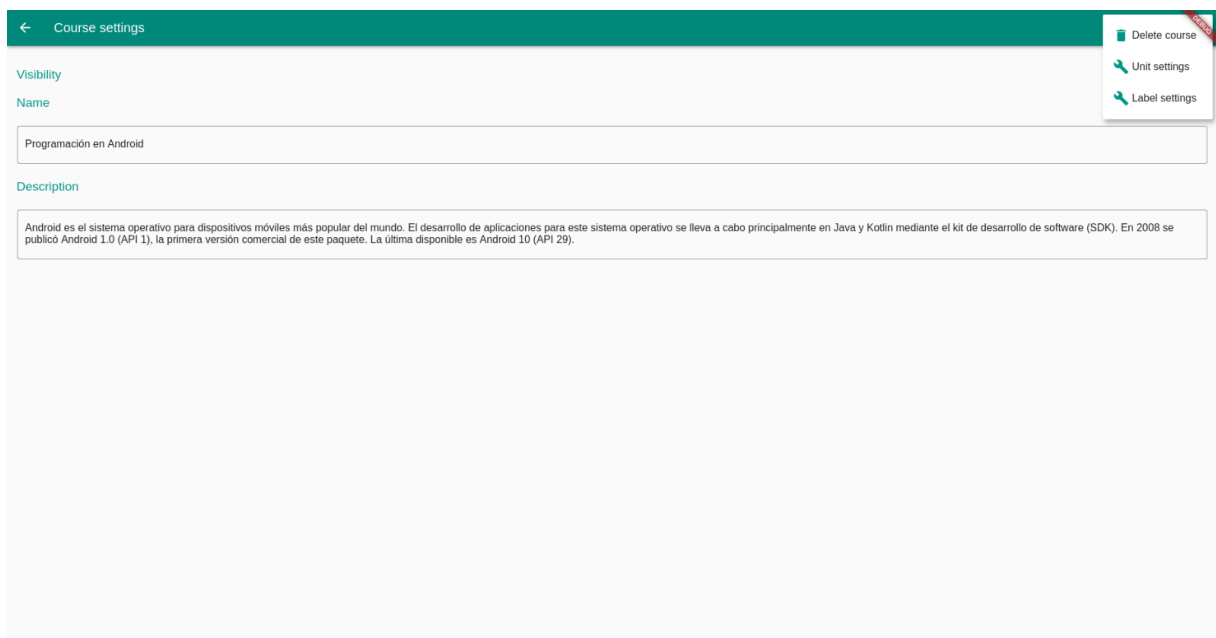


Figura F.13: Vista de ajuste de curso Web.



Figura F.14: Vista de ajuste de unidades Web.

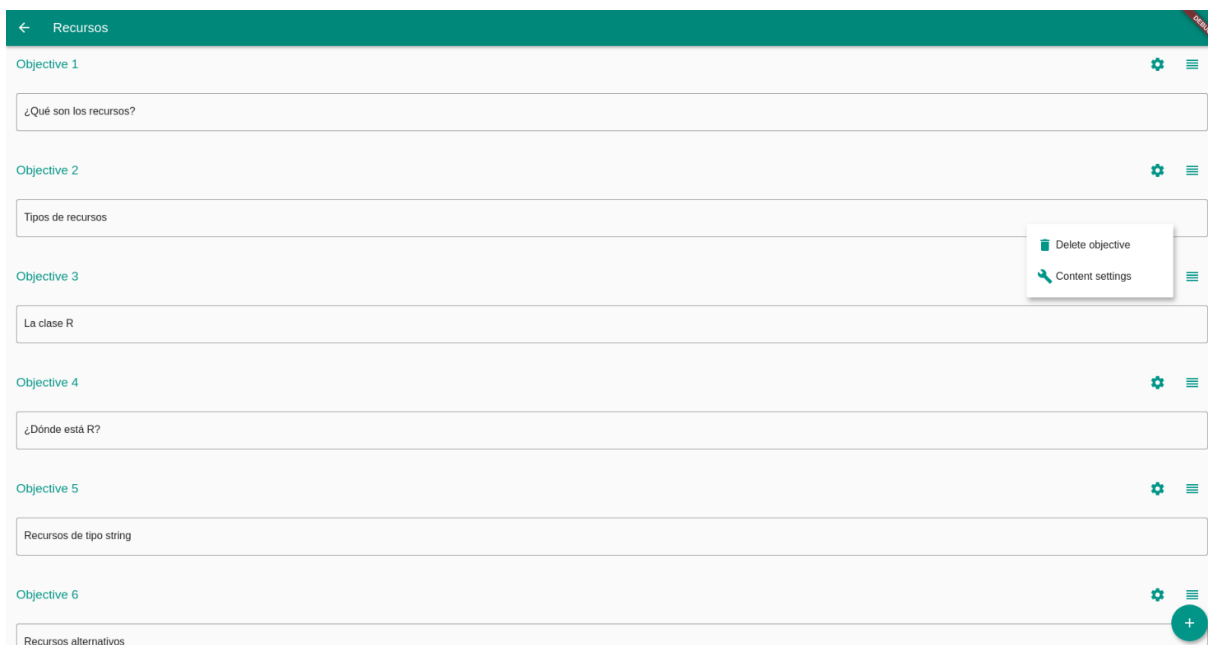


Figura F.15: Vista de ajuste de objetivos Web.

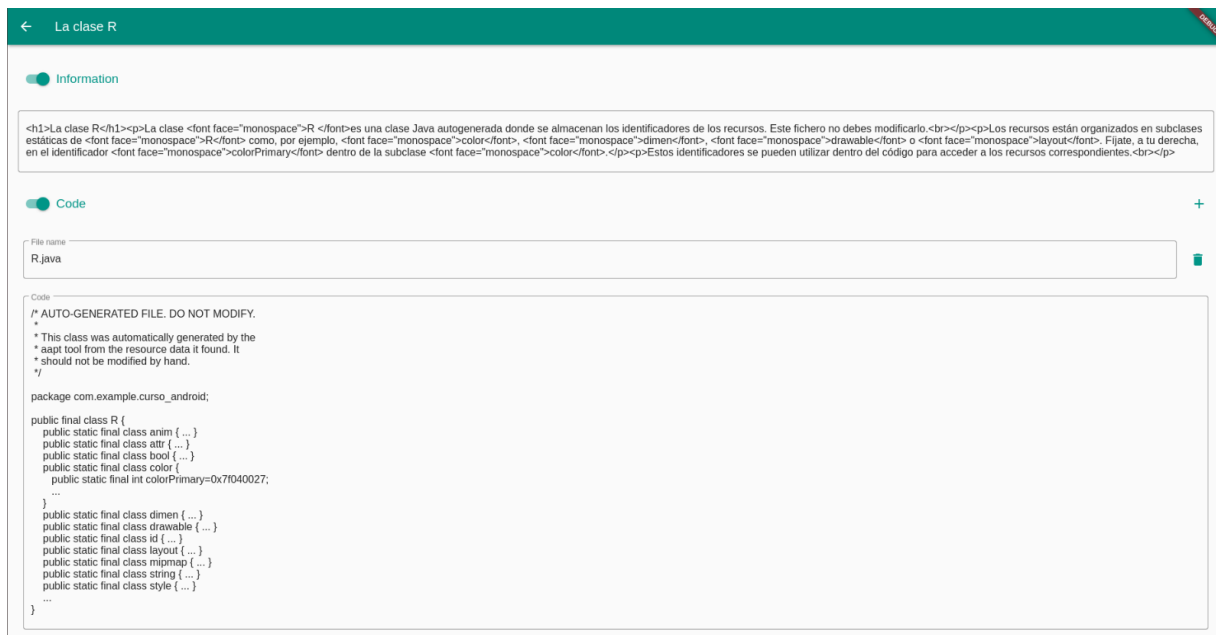


Figura F.16: Vista de ajuste de contenido: Información y código Web.

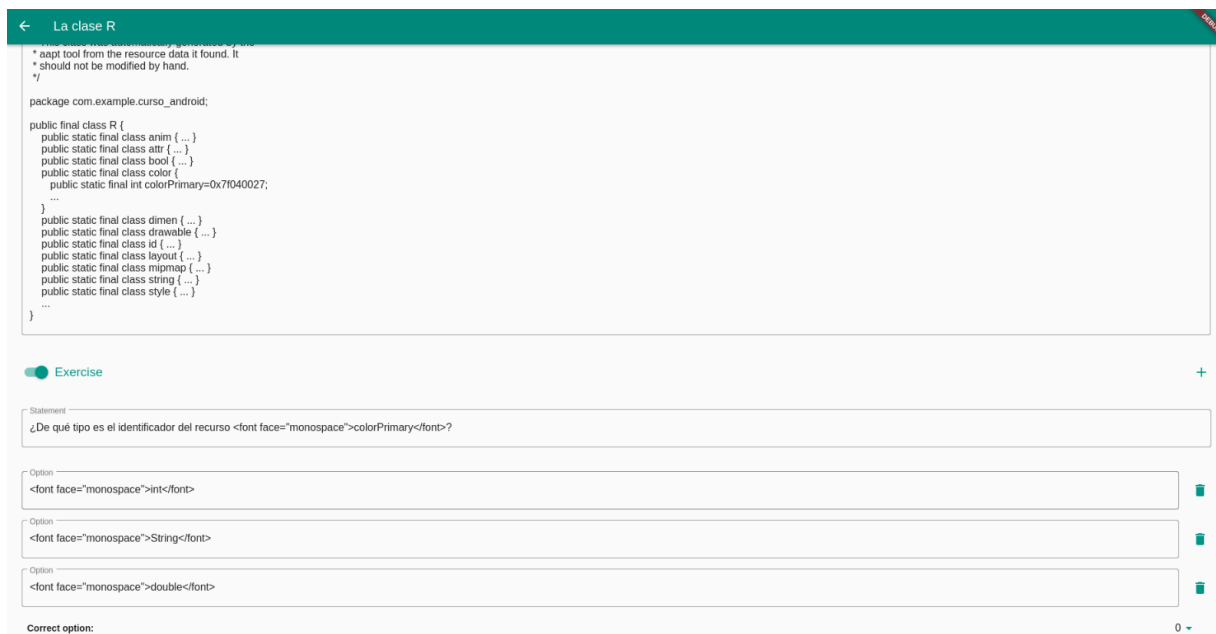


Figura F.17: Vista de ajuste de contenido: Ejercicio Web.

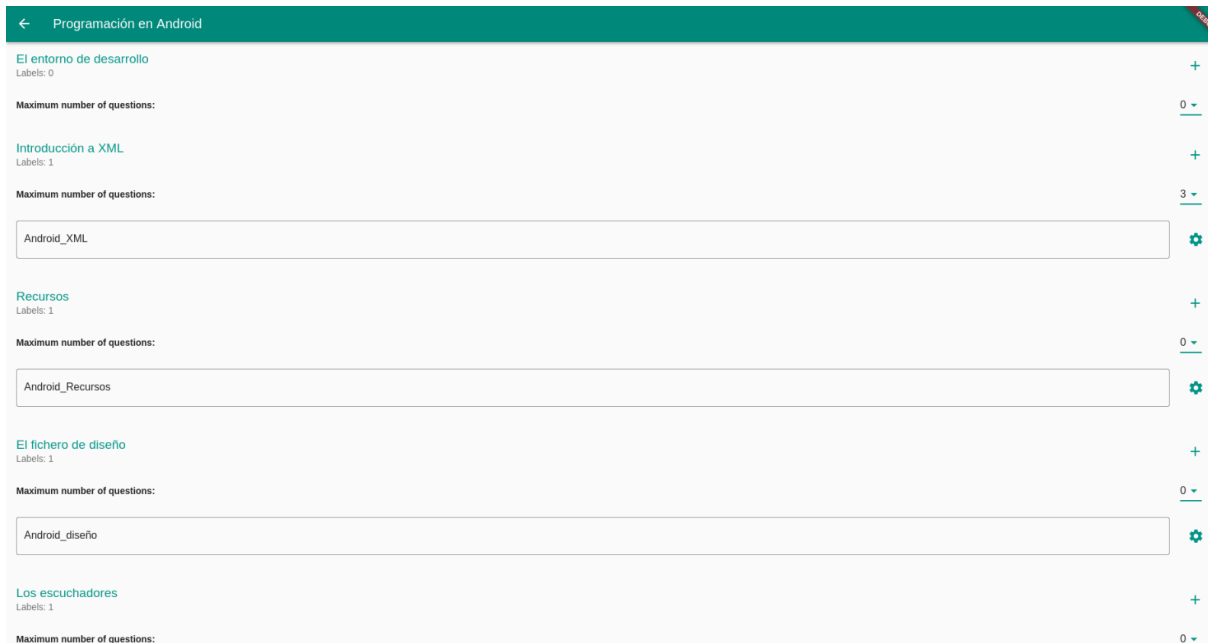


Figura F.18: Vista de ajuste de etiquetas Web.

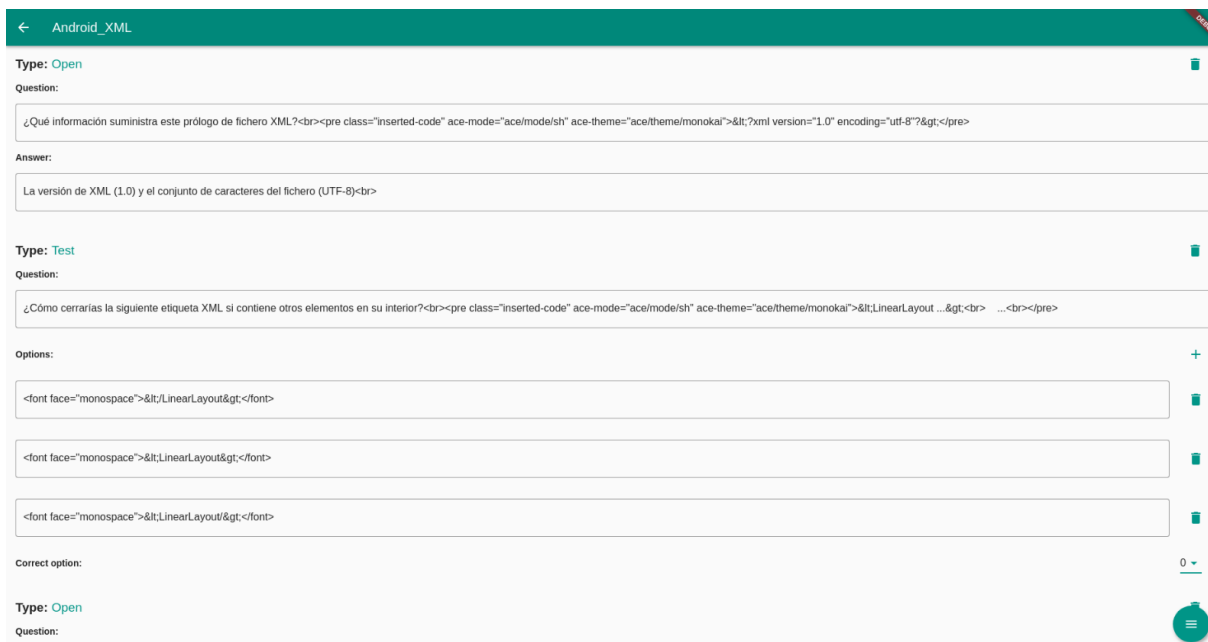


Figura F.19: Vista de ajuste de preguntas Web.

F.2. Vista móvil

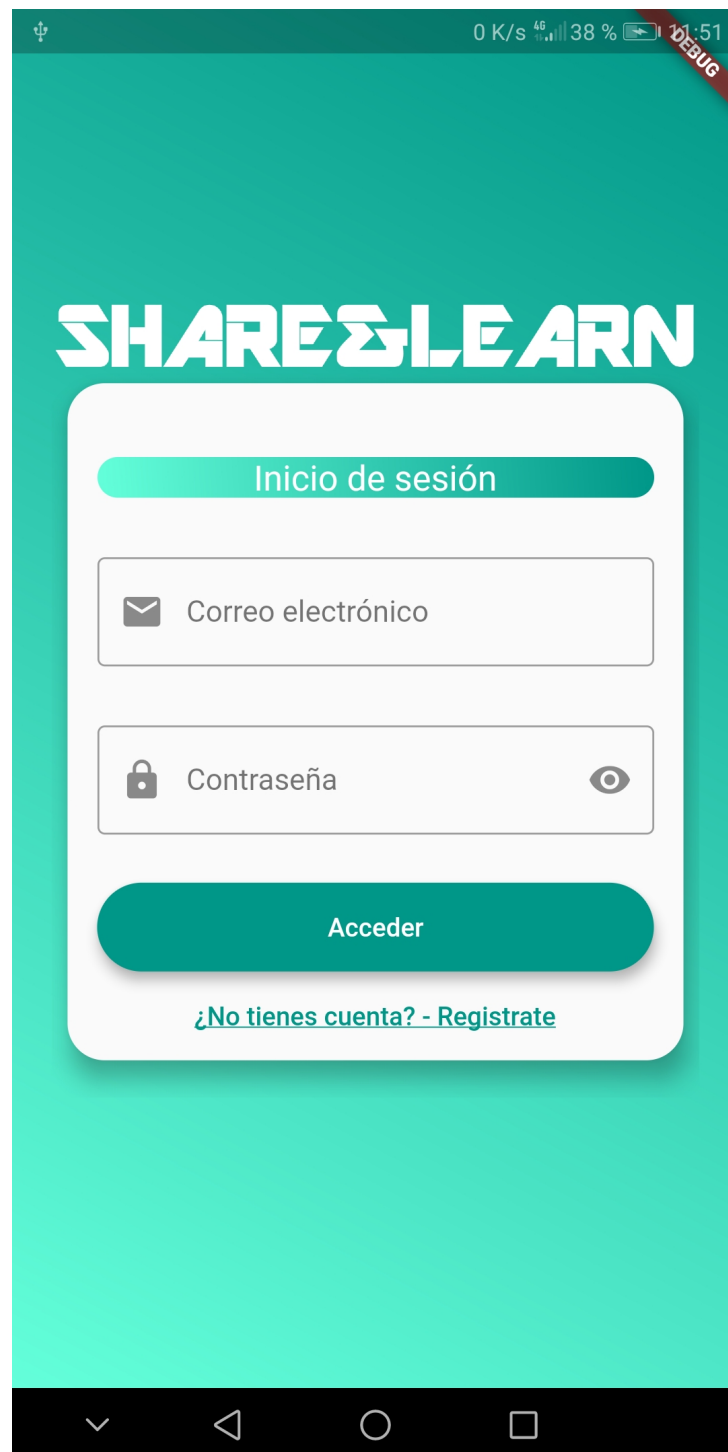


Figura F.20: Vista de inicio de sesión móvil.

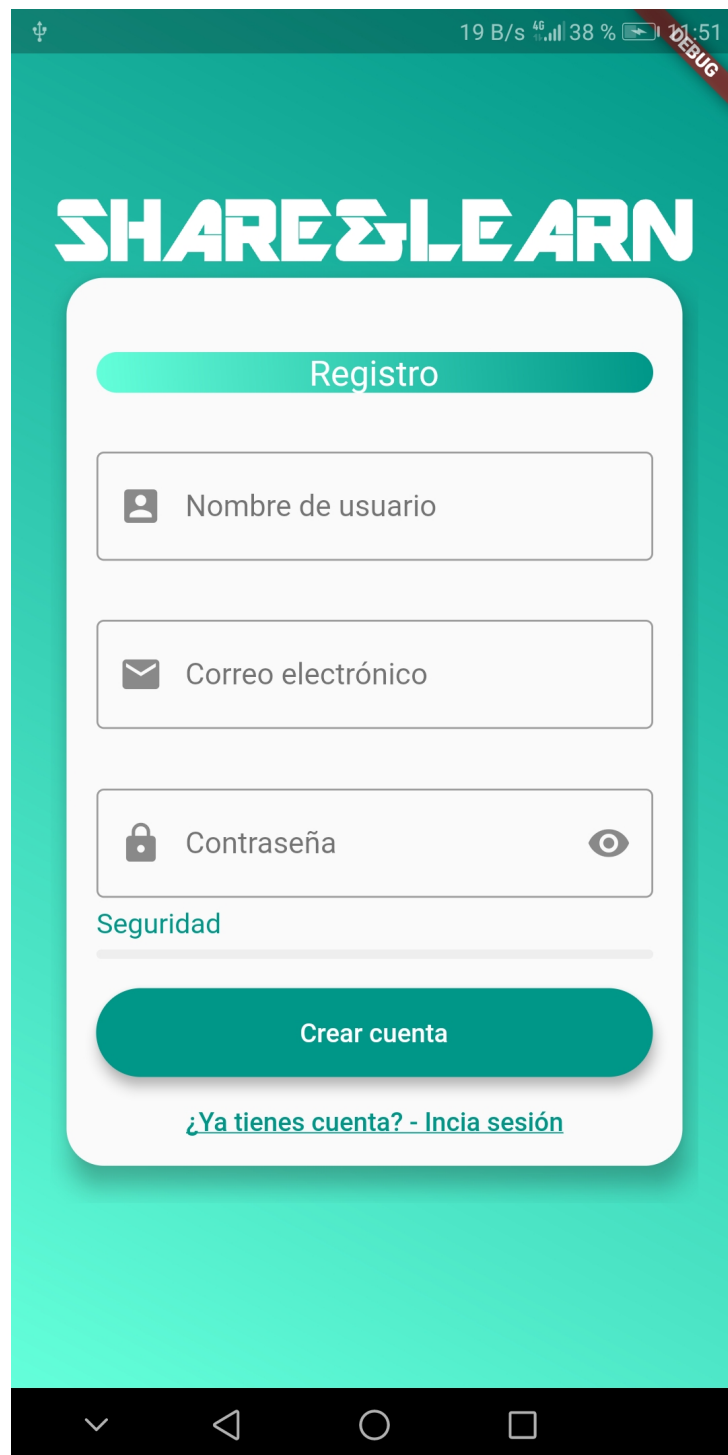


Figura F.21: Vista de registro de usuario móvil.

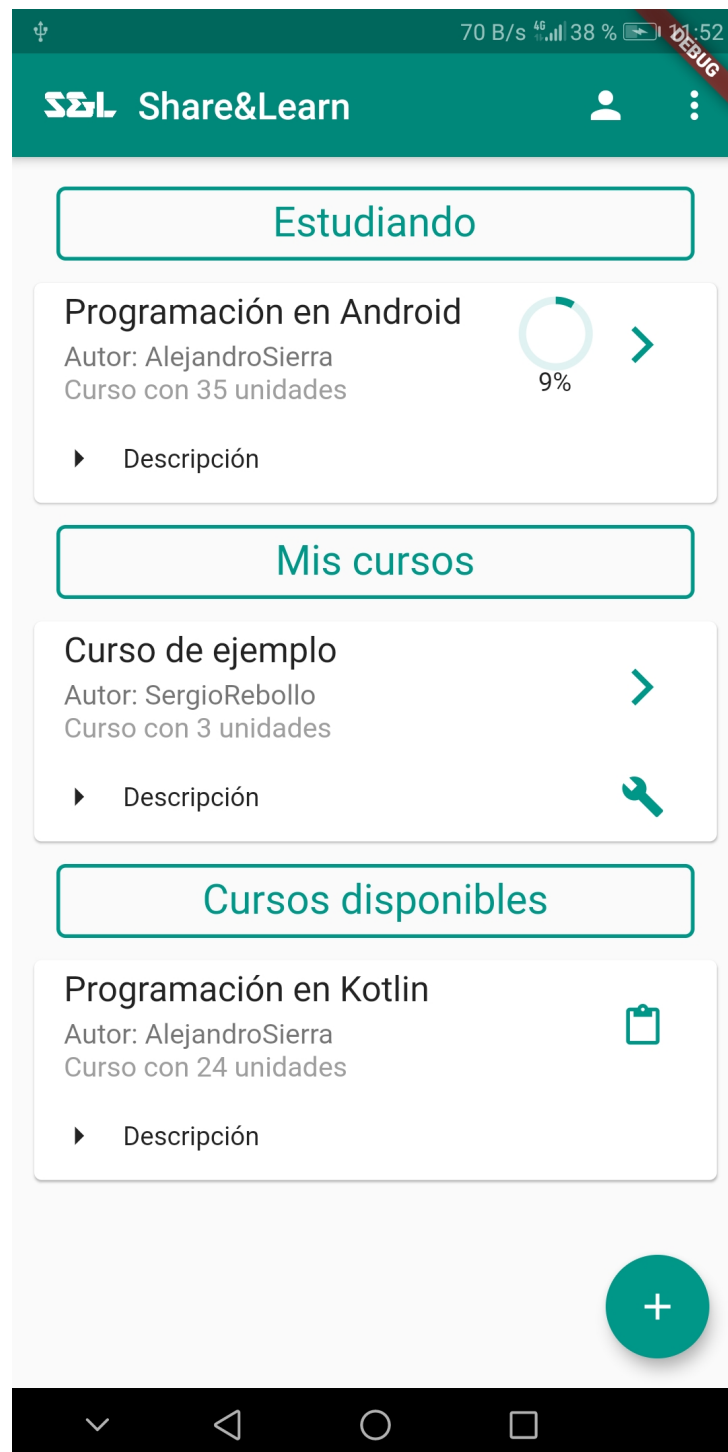


Figura F.22: Vista de pantalla principal móvil.

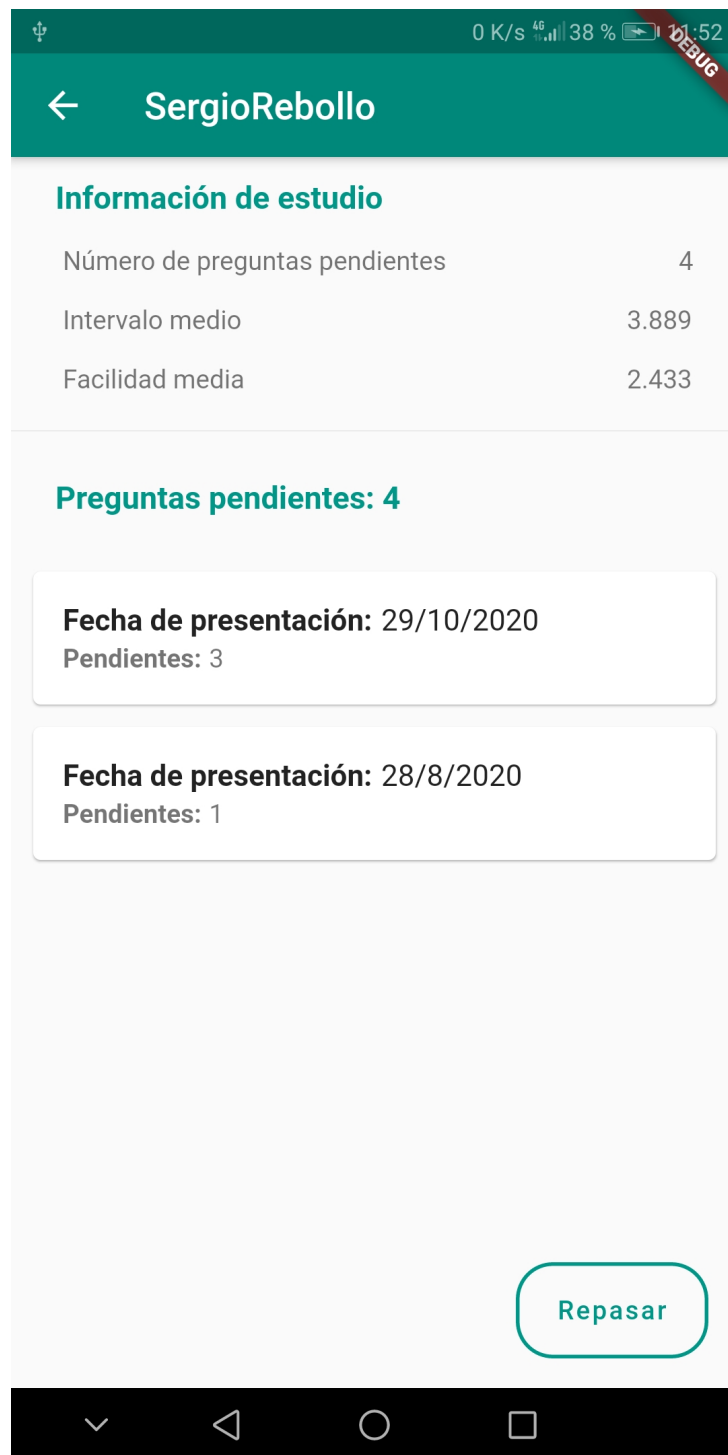


Figura F.23: Vista de seguimiento de usuario móvil.



Figura F.24: Vista de pregunta abierta móvil.

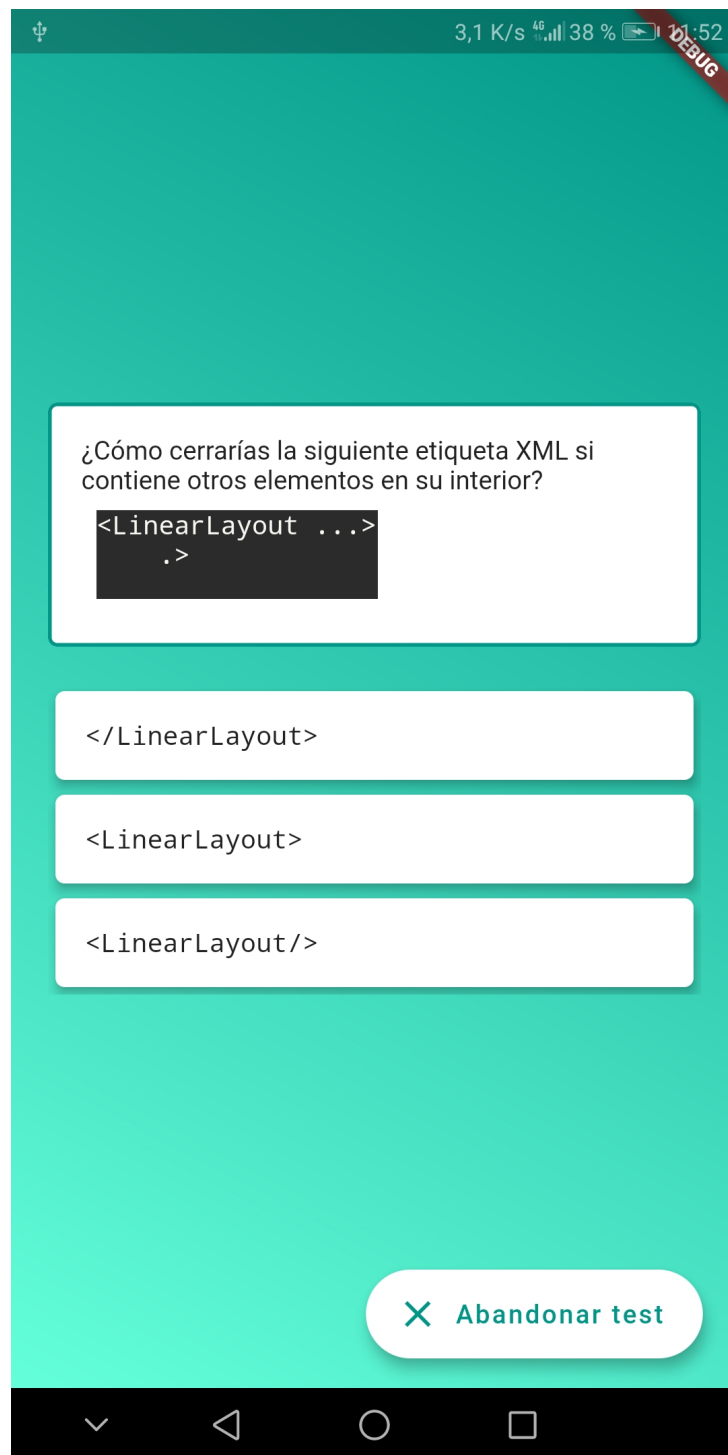


Figura F.25: Vista de pregunta tipo test móvil.

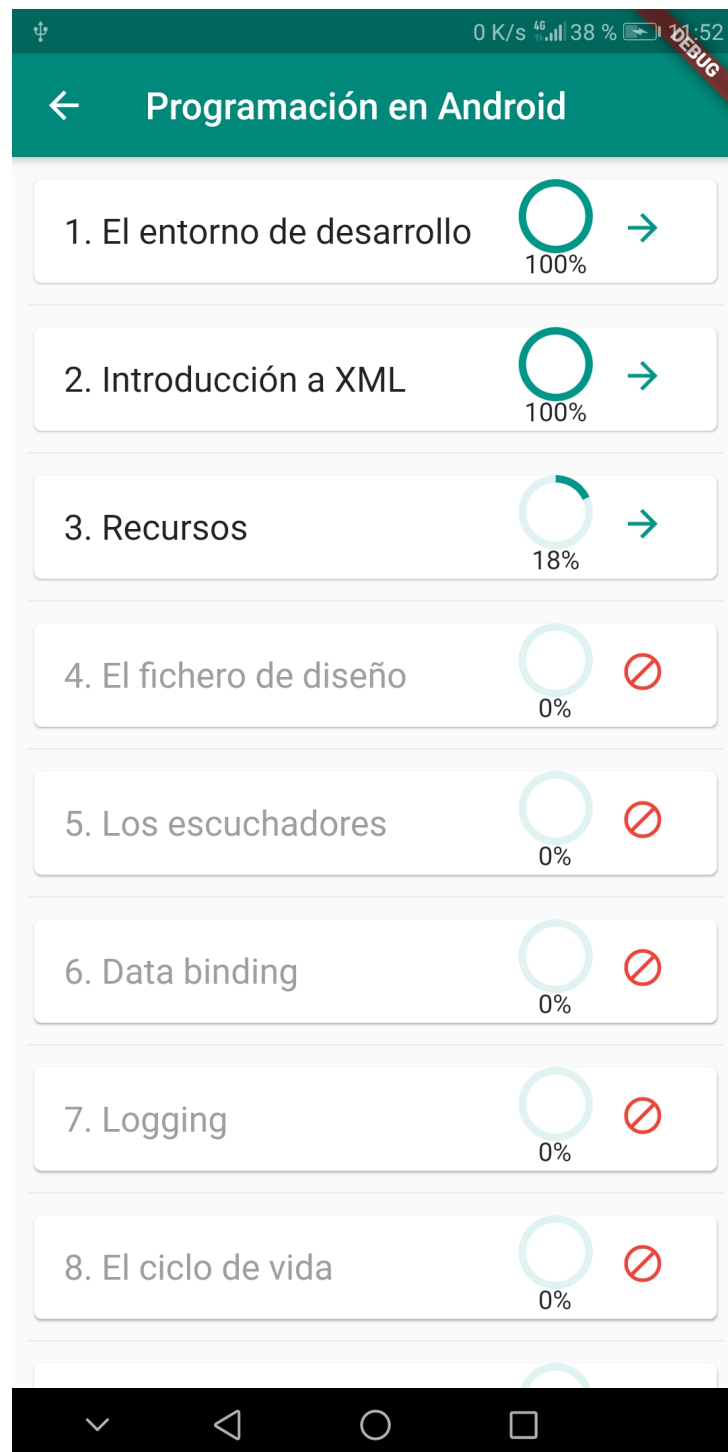


Figura F.26: Vista de lista de unidades móvil.

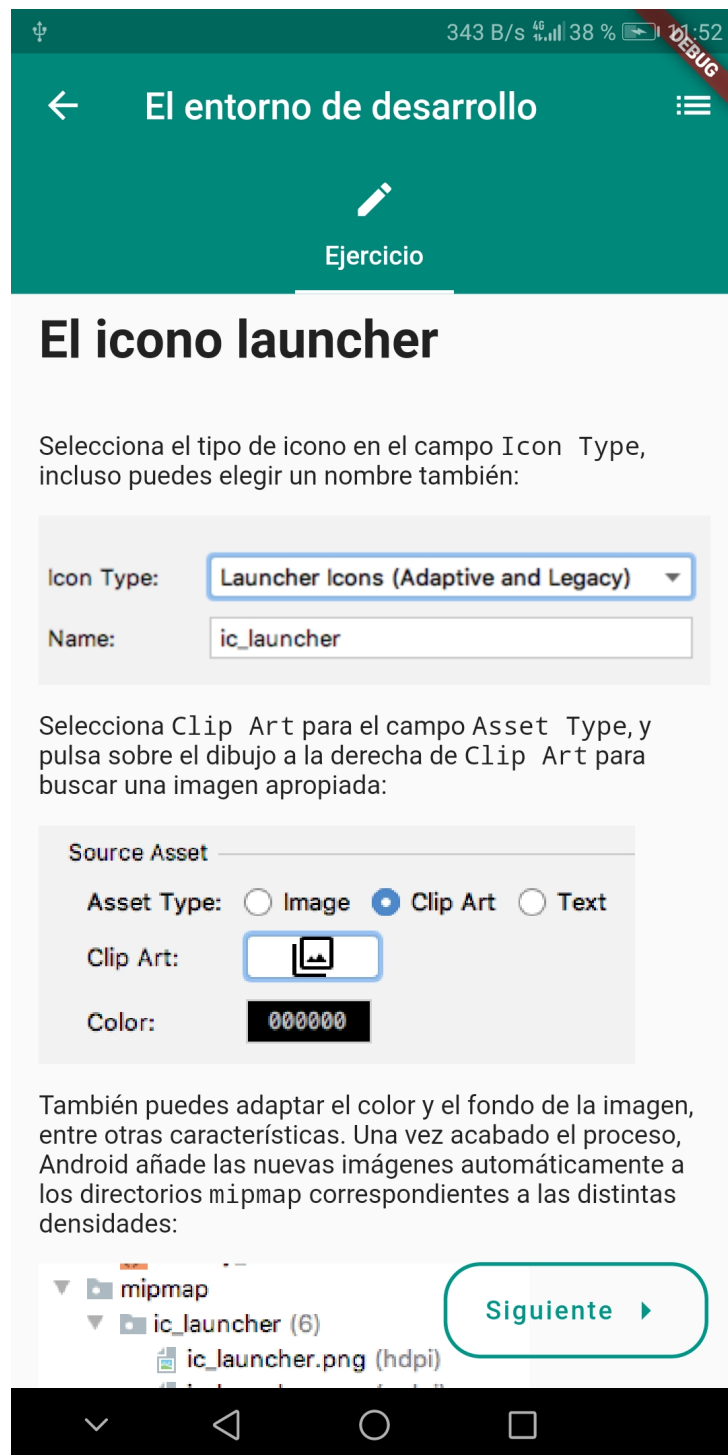


Figura F.27: Vista de ejemplo de objetivo 1 móvil.

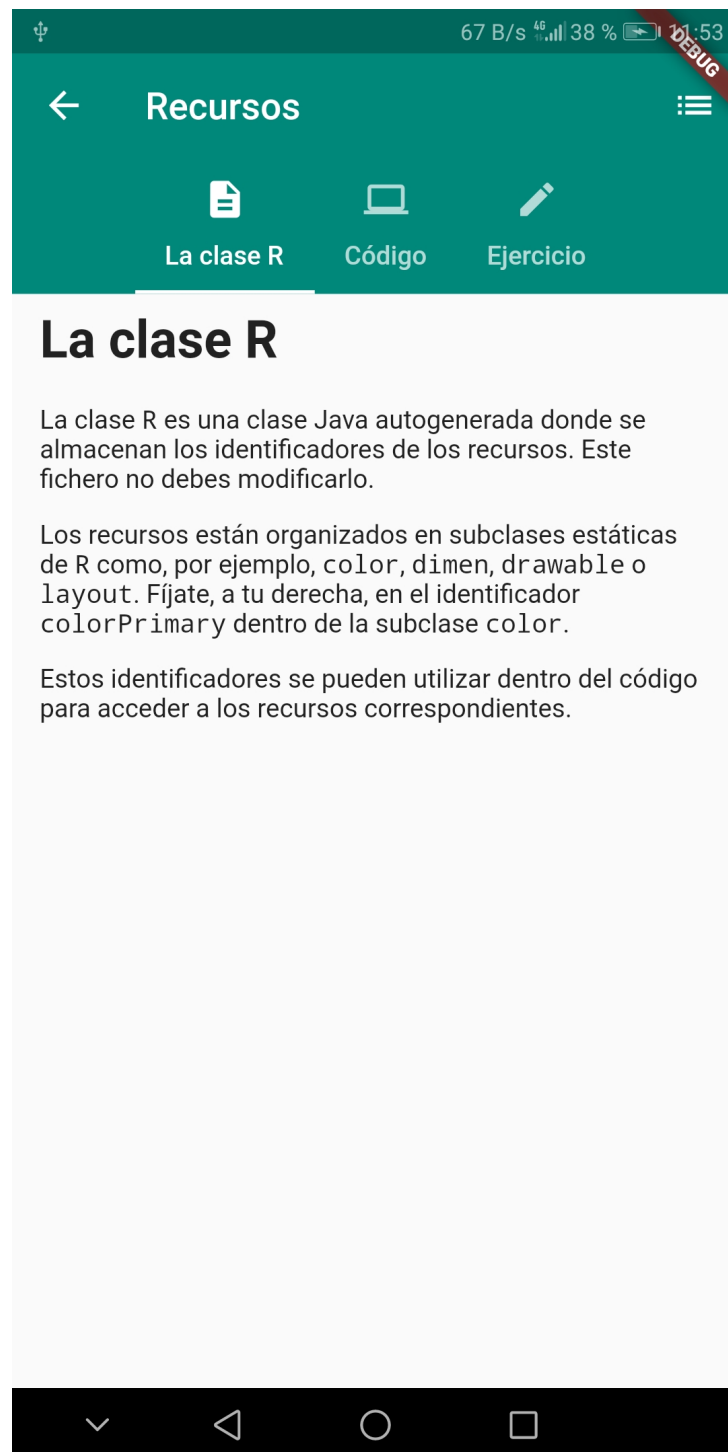


Figura F.28: Vista de ejemplo de objetivo 2 móvil.

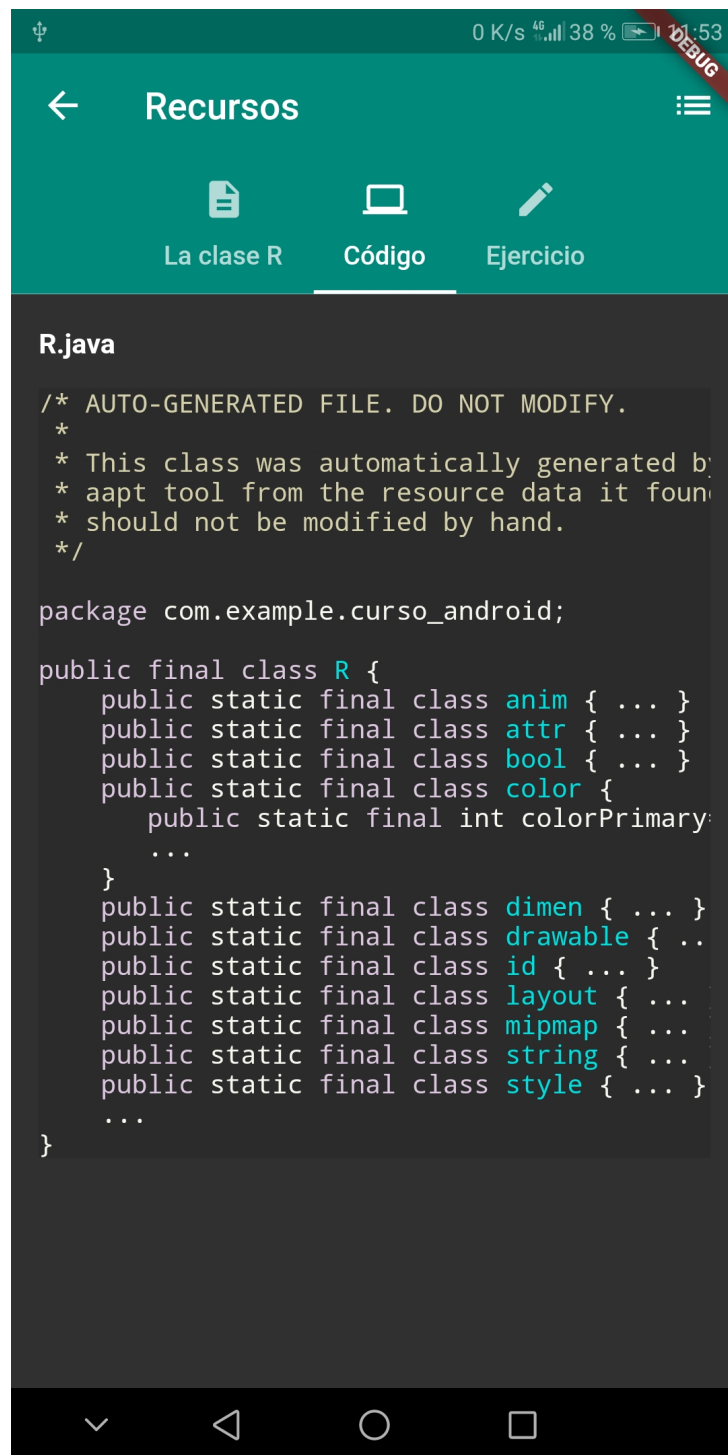


Figura F.29: Vista de ejemplo de objetivo 3 móvil.

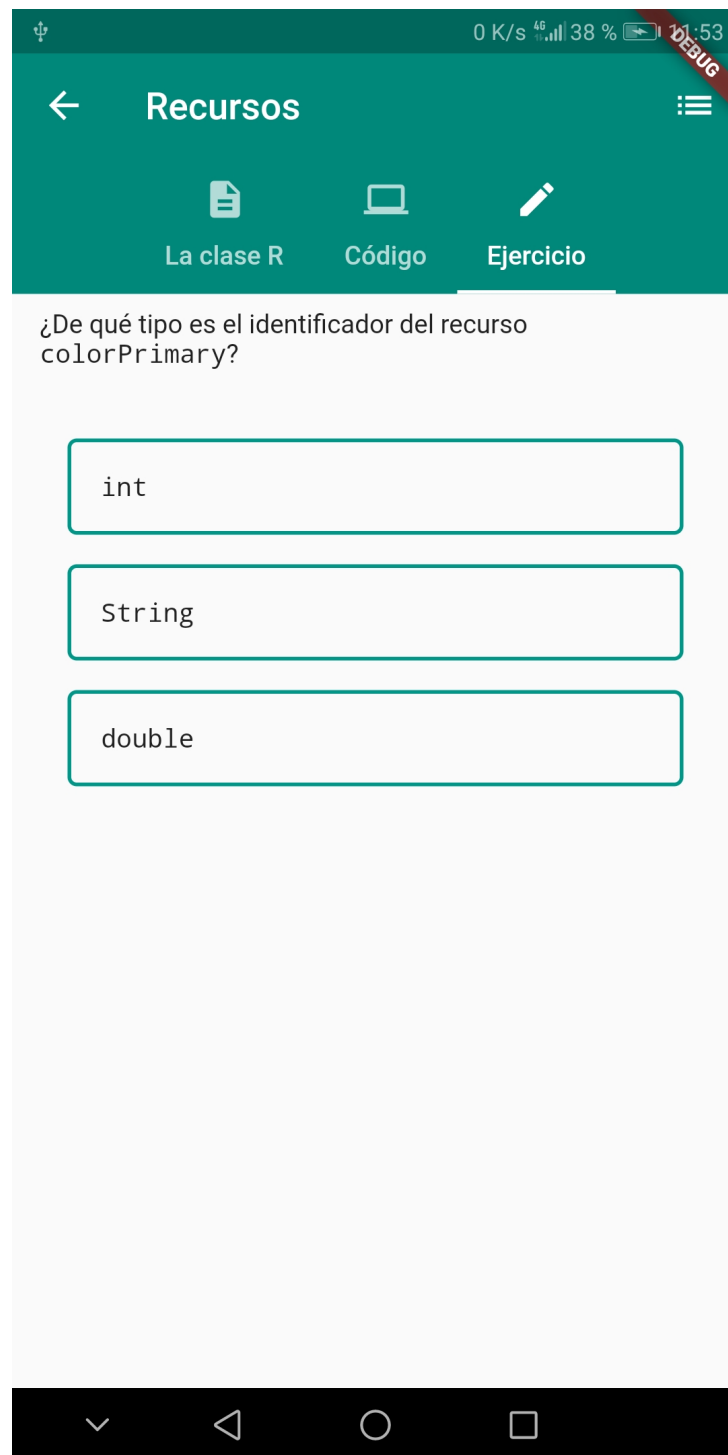


Figura F.30: Vista de ejemplo de objetivo 4 móvil.

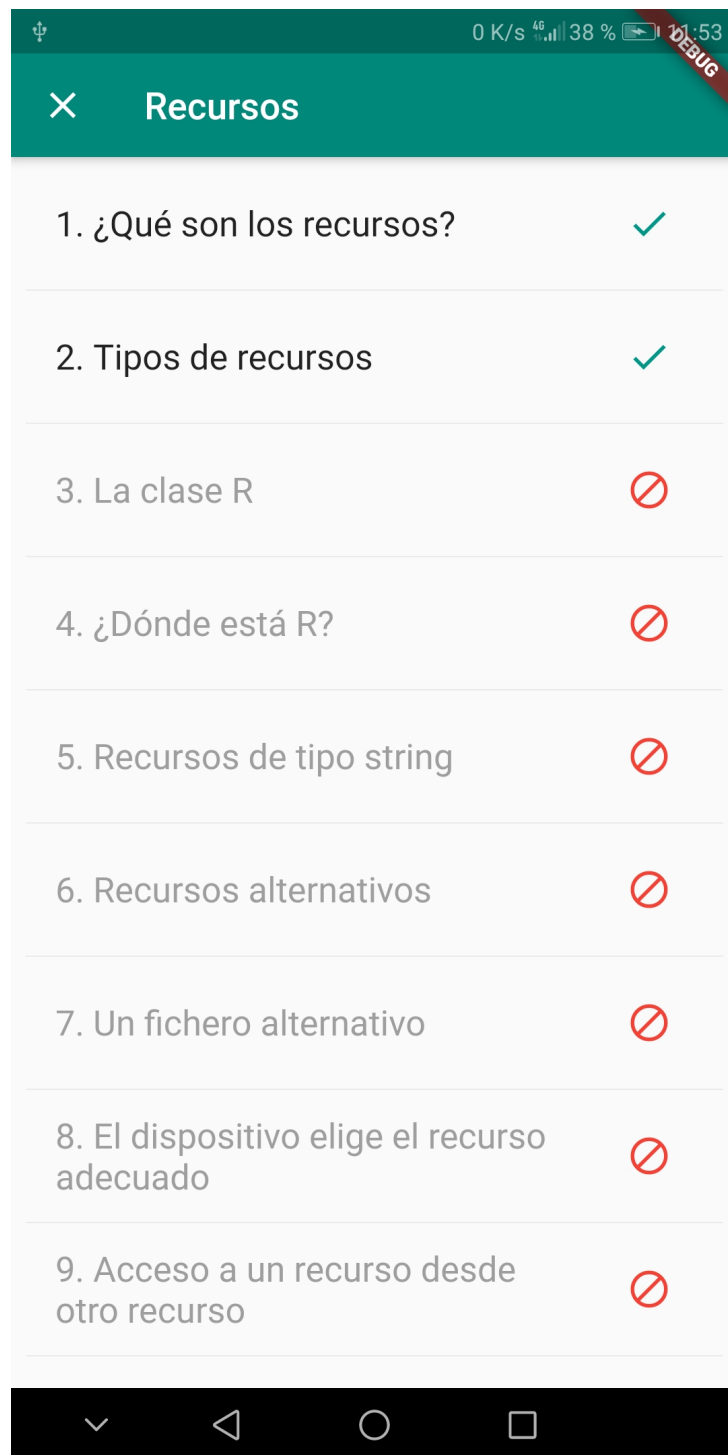


Figura F.31: Vista de lista de objetivos móvil.

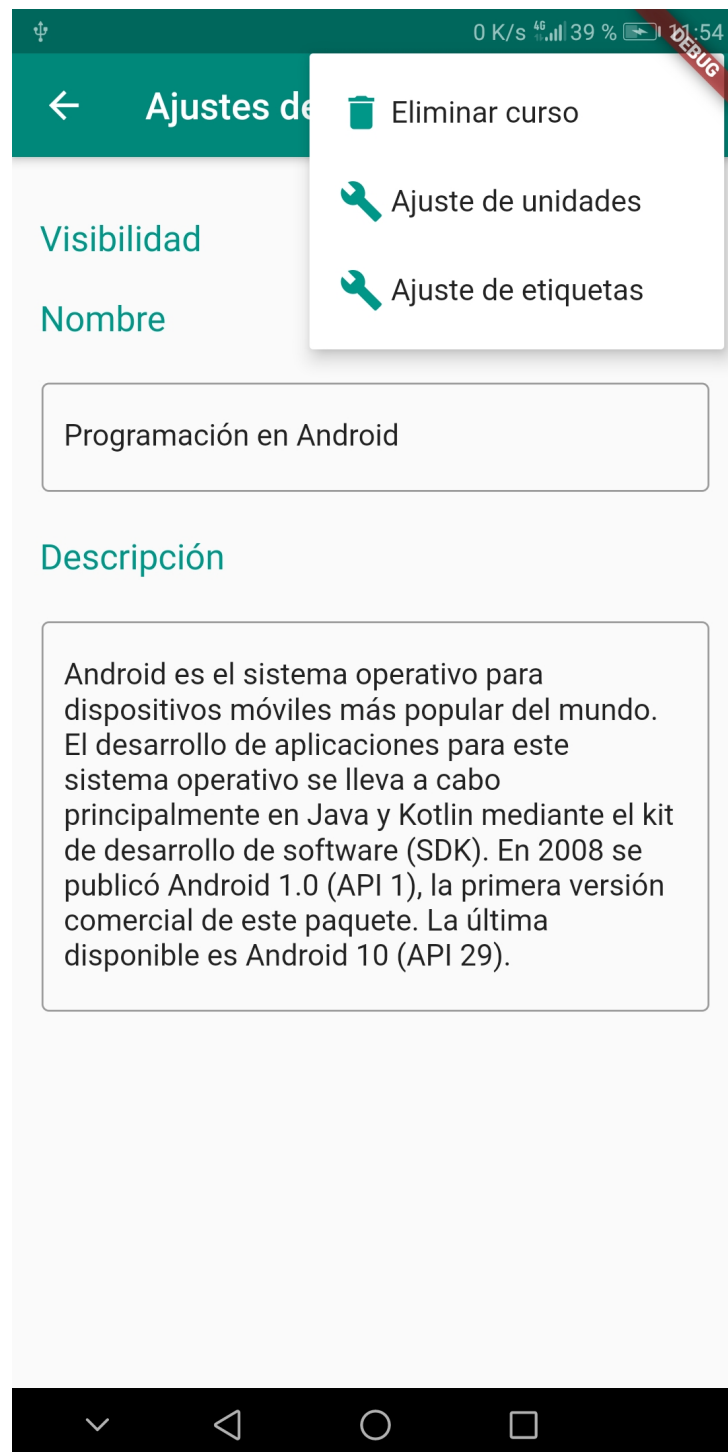


Figura F.32: Vista de ajustes de curso móvil.

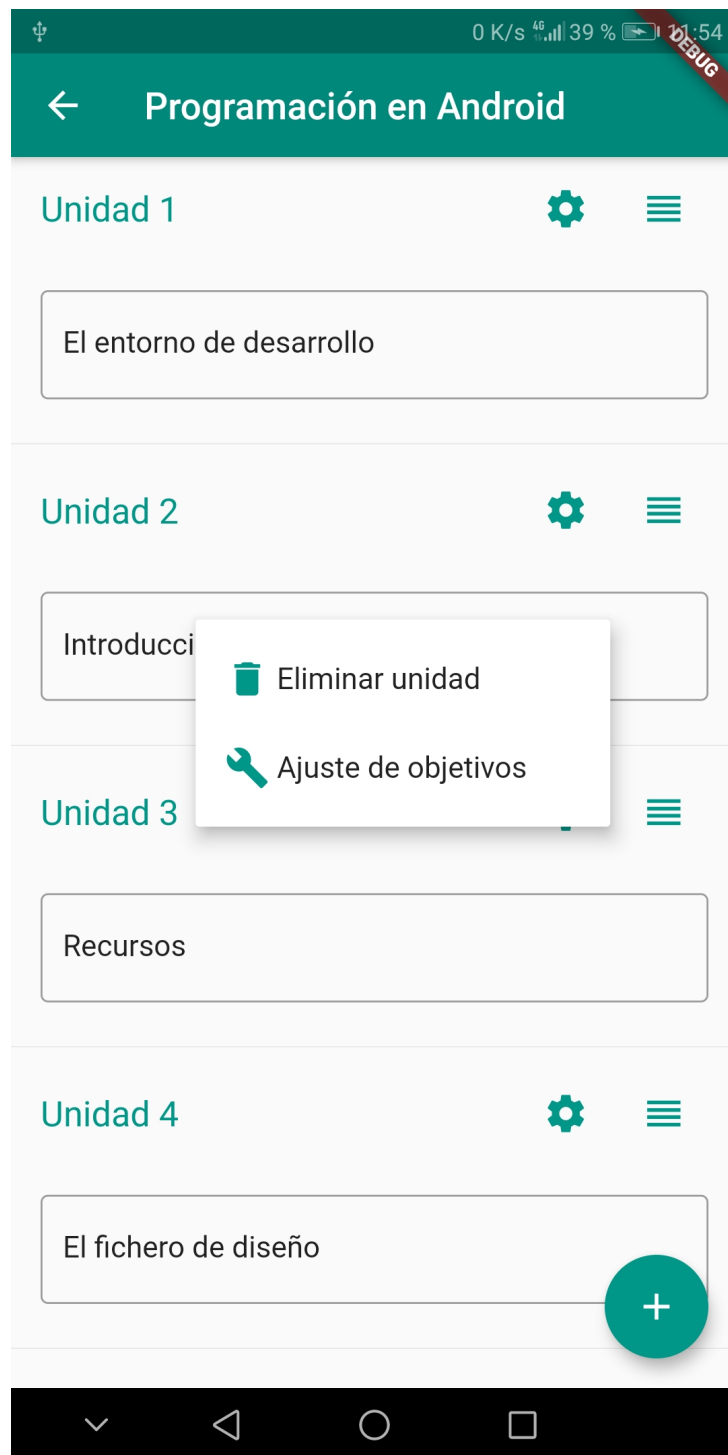


Figura F.33: Vista de ajustes de unidades móvil.

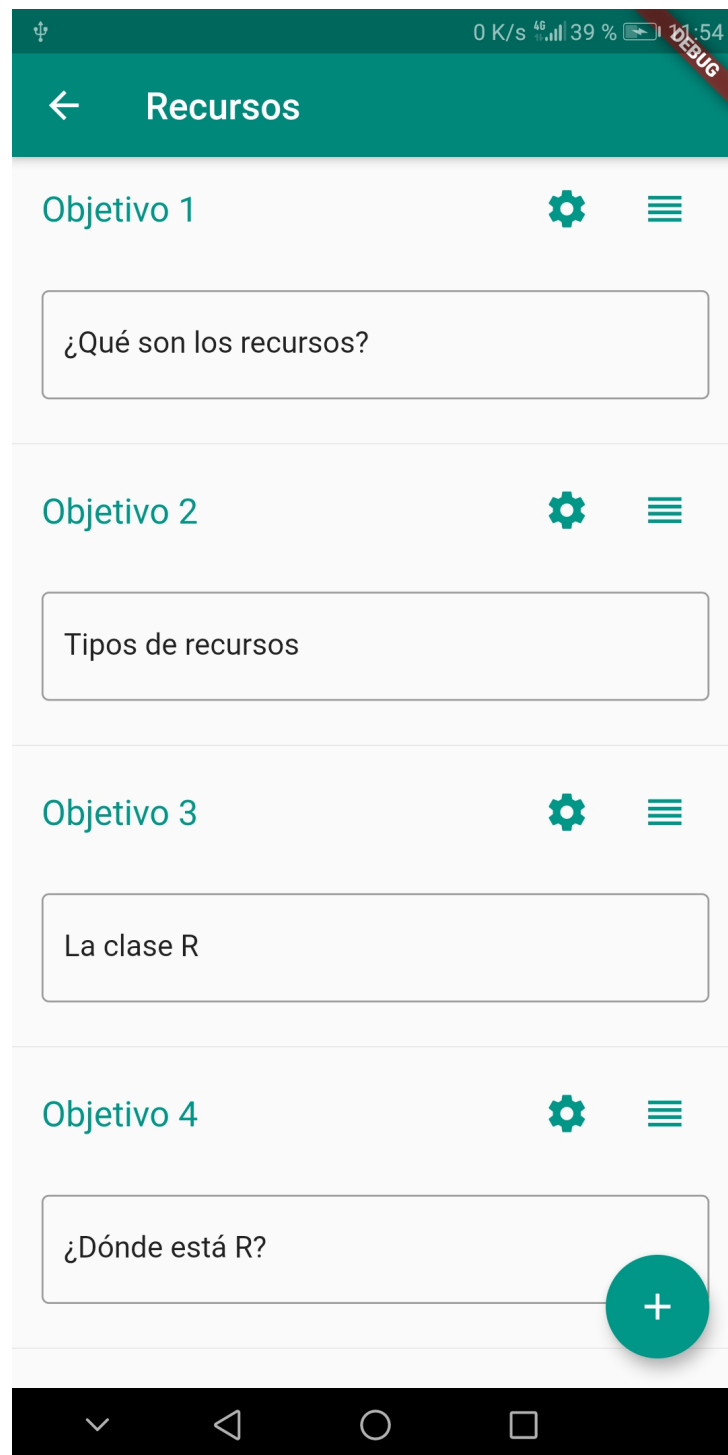


Figura F.34: Vista de ajustes de objetivos móvil.

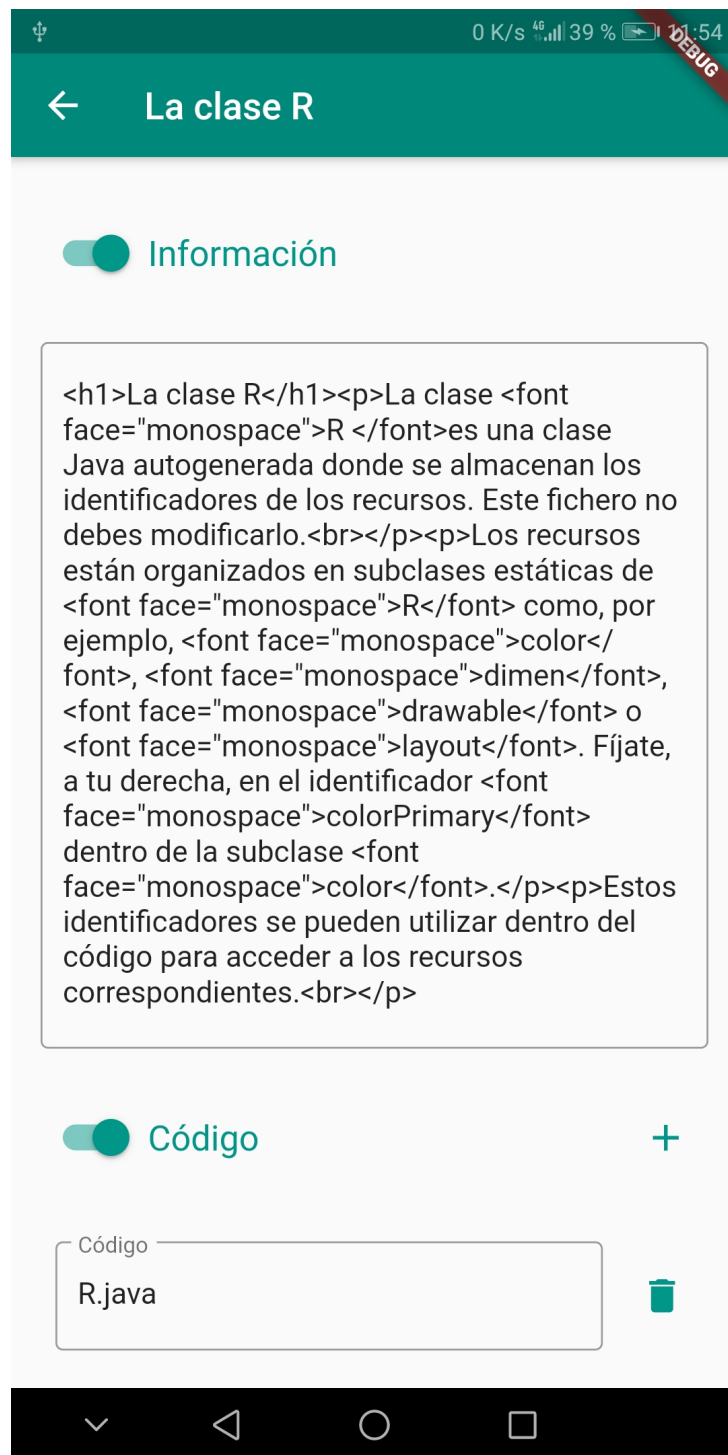


Figura F.35: Vista de ajustes de contenido: Información móvil.

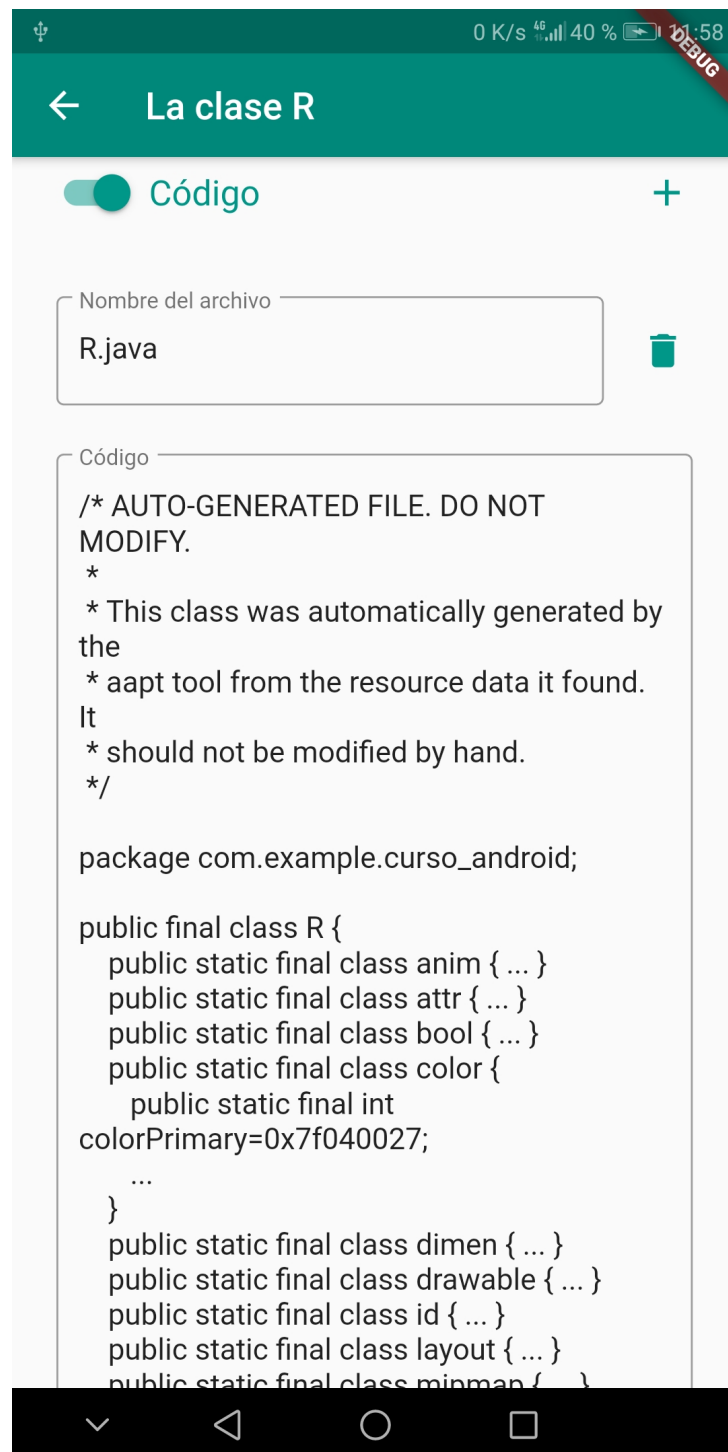


Figura F.36: Vista de ajustes de contenido: Código móvil.

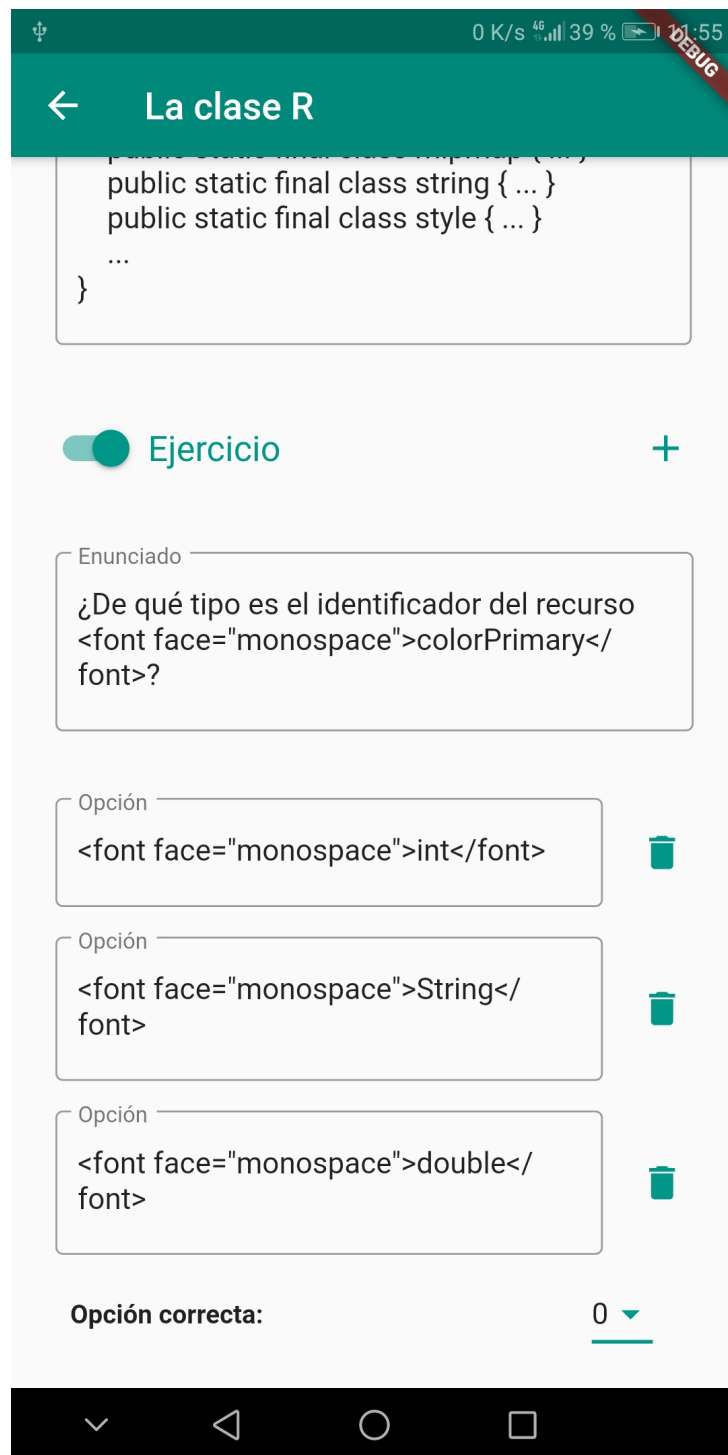


Figura F.37: Vista de ajustes de contenido: Ejercicio móvil.

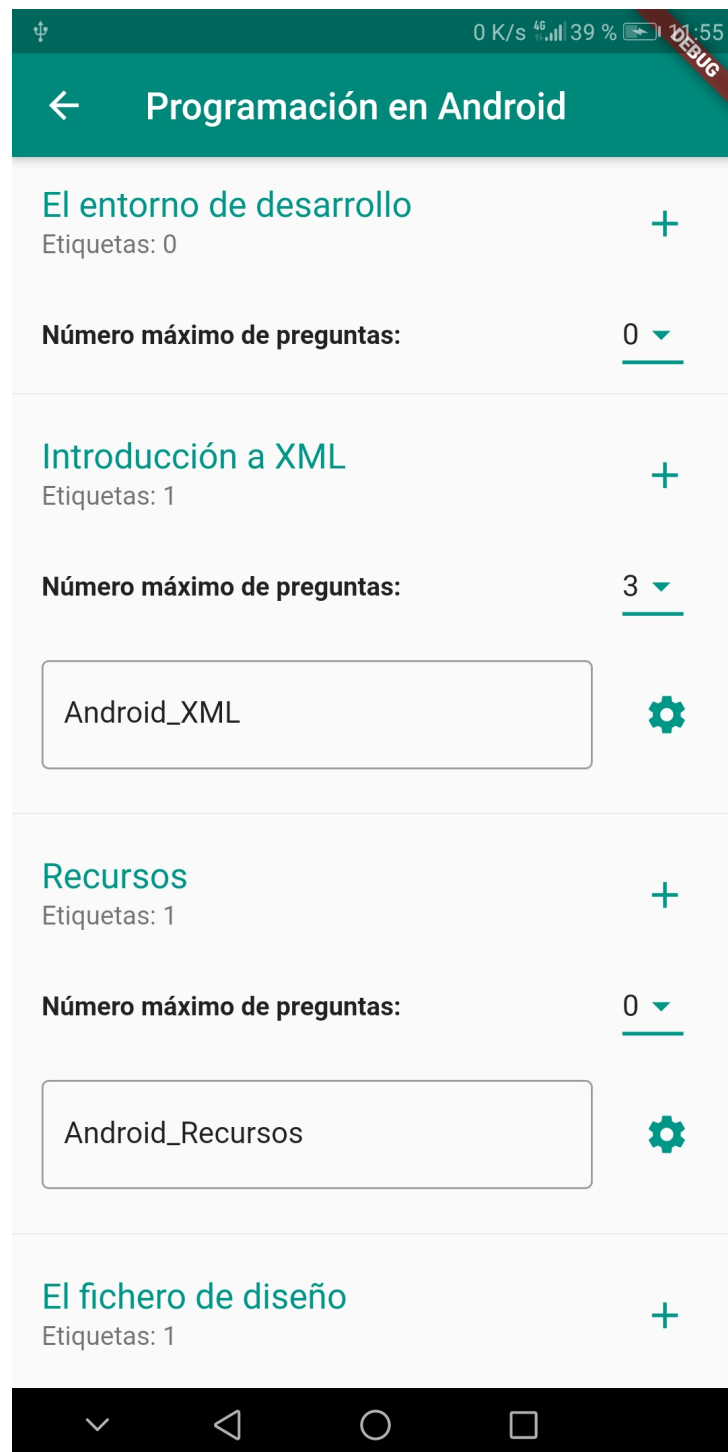


Figura F.38: Vista de ajustes de etiquetas móvil.

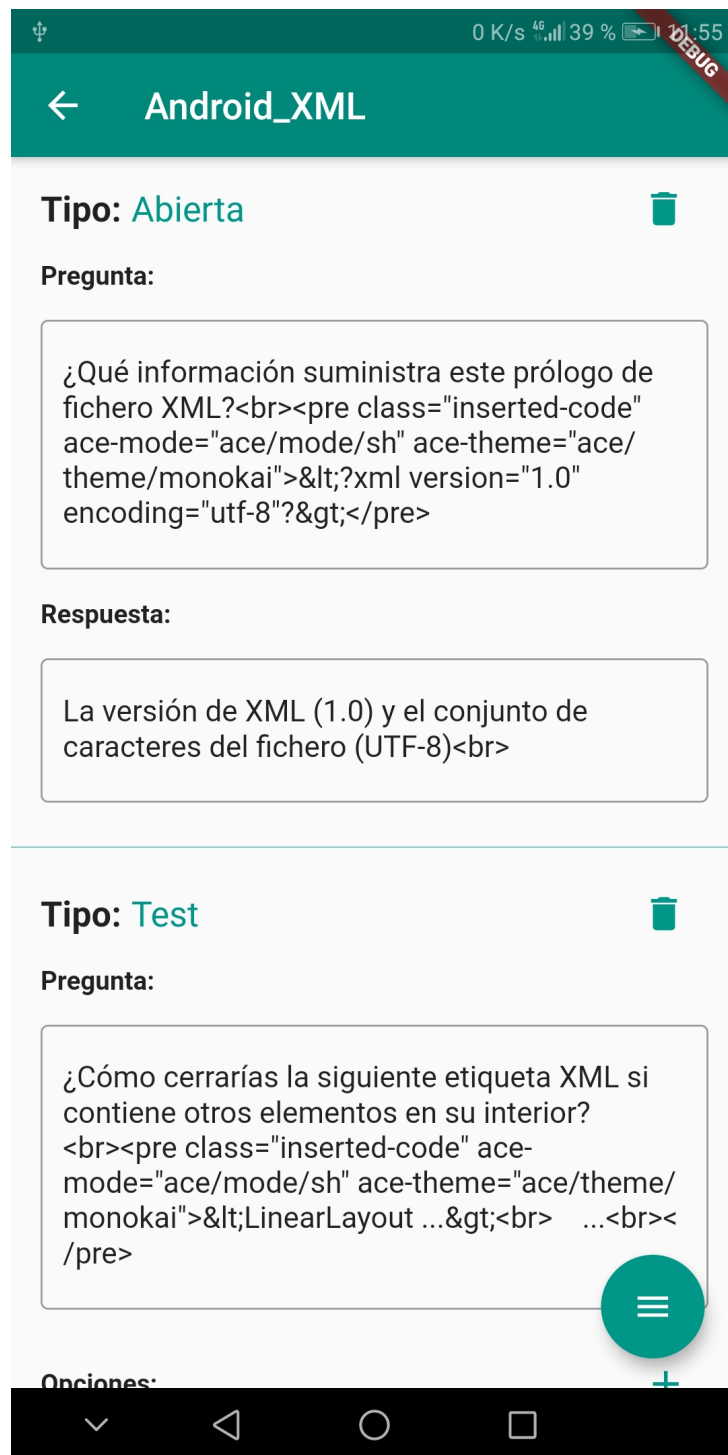


Figura F.39: Vista de ajustes de pregunta: Abierta móvil.

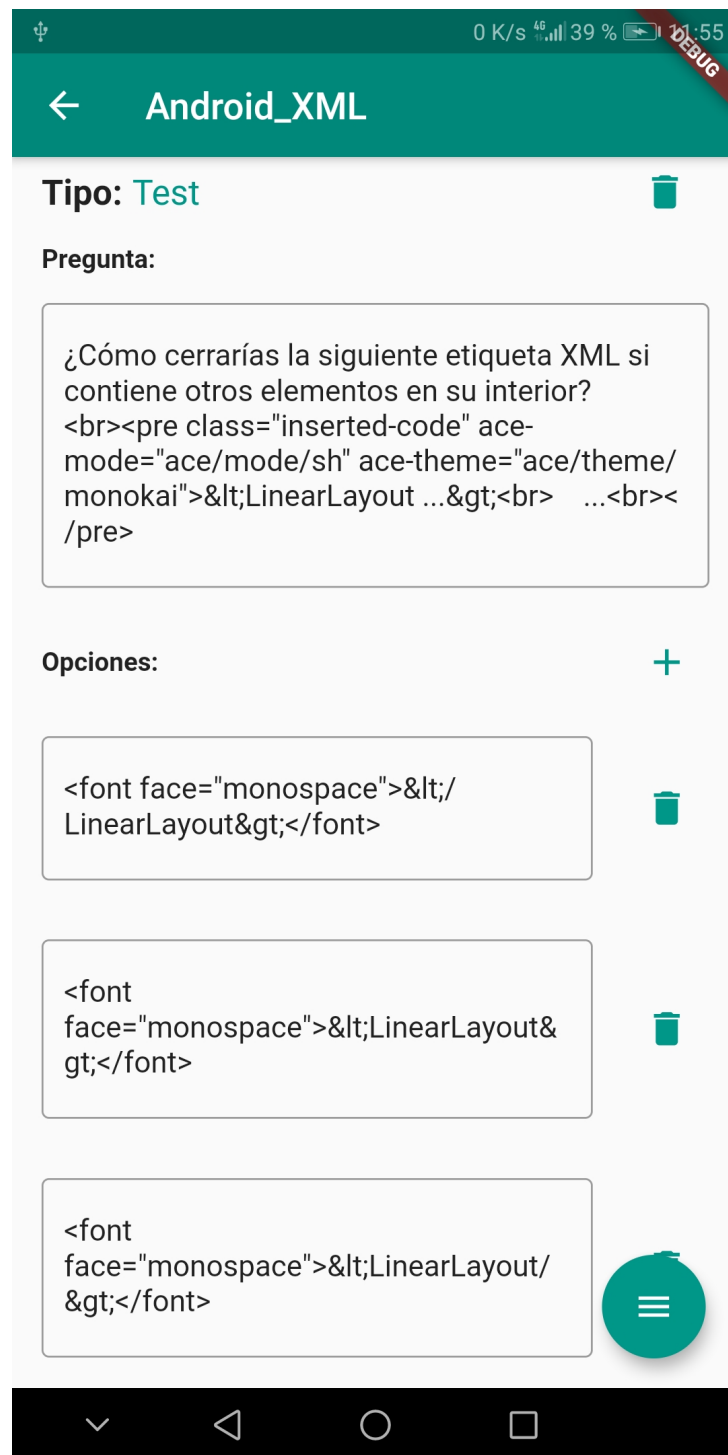


Figura F.40: Vista de ajustes de pregunta: Test móvil.